(2)

Research and Development Technical Report
SLCET-TR-88-0376-8

Very High Speed Integrated Circuits (VHSIC) Phase 2 Submicrometer Technology Development

INTEROPERABILITY STANDARDS

Dale Rickard (IBM)
George Anderson (Honeywell)
George McIver (TRW)

IBM Federal Systems
Systems Integration Division
9500 Godwin Drive
Manassas, VA 22110

November 1988

DISTRIBUTION STATEMENT

Approved for public release;
distribution is unlimited.

Prepared for
ELECTRONICS TECHNOLOGY AND DEVICES LABORATORY

US ARMY
LABORATORY COMMAND
FORT MONMOUTH, NEW JERSEY 07703-5000

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) SLCET-TR-85-0376-8
6a. NAME OF PERFORMING ORGANIZATION IBM Federal ~~Systems~~	6b. OFFICE SYMBOL (If applicable)
6c. ADDRESS (City, State, and ZIP Code) Systems Integration Division 9500 Godwin Drive Manassas, VA 22110	7a. NAME OF MONITORING ORGANIZATION US Army Laboratory Command (LABCOM) Electronics Technology & Devices Lab
	7b. ADDRESS (City, State, and ZIP Code) ATTN: SLCET-IV Fort Monmouth, NJ 07703-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION VHSIC Program Office	8b. OFFICE SYMBOL (If applicable) RM 3D-139
	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAK20-85-C-0376
8c. ADDRESS (City, State, and ZIP Code) Pentagon Washington, DC	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
63452F	2700		DA313825

11. TITLE (Include Security Classification)
Very High Speed Integrated Circuits (VHSIC) Phase 2 Submicrometer Technology Development, Interoperability Standards (U)

12. PERSONAL AUTHOR(S)
Dale Rickard, George Anderson, George McIver (Contd)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interoperability Stds	FROM 10Oct85 TO 310ct88	88 November 30	257

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
09	01		VHSIC, Submicrometer integrated Circuits, Integrated Circuit Fabrication, CMOS Integrated Circuits, Multi-Chip Integrated Circuit Packaging
09	07		

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report covers the Interoperability Standards developed under the VHSIC Phase 2 Submicrometer Technology Development Program. The Interoperability Standards are the result of joint effort by the three VHSIC Phase 2 contractors and representatives of the Army, Air Force and Navy to develop standards for the interconnection and operation of integrated circuits fabricated with 0.5 micron technology. An introductory description and a copy of each standard is provided. The standards developed are:

a. The Pi-bus standard interconnect system bus,
b. The TM-bus standard system maintenance bus,
c. The ETM-bus standard chip level maintenance bus, and
d. Electrical and Clocking Interface to VHSIC Chips.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
VINCENT J. ORGANIC	201-544-3477	SLCET-IV

DD Form 1473, JUN 86 *Previous editions are obsolete.*

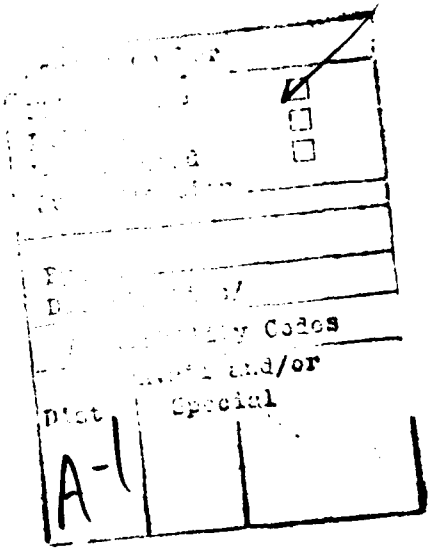
12. Personal Author(s) Contd)

The VHSIC Interoperability Standards were developed as a joint effort of the three Phase 2 Submicrometer Technology Development contractors and representatives of the Army, Air Force and Navy. The report authors, contractors and contract numbers are:

Dale Richard
IBM Corporation
Federal Systems
9500 Godwin Drive
Manassas, VA 22110
Contract No. DAAK20-85-C-0376

George Anderson
Honeywell Corporation
Solid State Electronics Division
12001 State Highway 55
Plymouth, MN 55441
Contract No. F33615-84-C-1500

George McIver
TRW Corporation
Electronics and Technology Division
One Space Park
Redondo Beach, CA 90278
Contract No. N00039-85-C-0111



The copyright release notices on the cover pages of Appendixes B and E are valid.
Per Mr. Robert Sproat, VHSIC Program Office

CONTENTS

ILLUSTRATIONS

PREFACE

This report represents the Contract Data Requirements List (CDRL) item A011 for work being performed under VHSIC Phase 2 Contract Number DAAK20-85-C-0367. The work reported herein is in response to Statement of Work paragraph 3.5.1, "Interoperability Standards". A significant portion of this work was performed jointly by the primary VHSIC Phase 2 contractors: IBM, Honeywell and TRW.

Specifications have been developed for:

- a standard interconnect system bus called the PI-bus (Appendix B),
- a standard system maintenance bus called the TM-bus (Appendix C),
- a standard chip level maintenance bus called the ETM-bus (Appendix D) and
- electrical and clocking interfaces to VHSIC chips (Appendix E).

IBM has developed a Bus Interface Unit (BIU) chip in 0.5 micrometer CMOS technology to implement the PI-bus standard. A chip called the Diagnostic and Maintenance (DxMD) controller has been developed to support the module level maintenance bus standard (TM-bus). The ETM-bus is supported by an On-chip Monitor (OCM) macro which is used on each VHSIC chip. These development activities are described in related technical reports (CDRL Items A005, A006 and A009).

Section 1

INTRODUCTION

The text of the VHSIC Statement of Work pertaining to the work reported herein is repeated below for reference:

> " 3.5.1 Interoperability Standards: Interface/interoperability standards shall be established by agreement among all VHSIC Phase 2 Submicrometer contractors and the Government COTR's to assure that all chips developed under the VHSIC Phase 2 Submicrometer program are interoperable, both electrically and physically. Standard voltage level(s) shall be established and utilized for all chips and input/output levels shall be equivalent for all chip interfaces, whether contained in a single or multi-chip package. A VHSIC halfmicrometer Bus Interface Unit (BIU) chip shall be developed to facilitate module interoperability with a standard interconnect system bus. The BIU and any other VHSIC chips developed under this Phase 2 VHSIC Submicrometer program shall interface directly to a standard system maintenance bus to be defined by agreement among all the VHSIC Phase 2 Submicrometer contractors and the Government COTR's. All these standards shall be documented and delivered."

The goal of the interoperability task is to develop a set of standards that will allow VHSIC technology from various contractors to be integrated into DoD systems in an efficient and cost effective manner.

To meet the interoperability goal, we have identified requirements for interoperability standards at both the chip and module levels. The chip level standards include clocks, power supply voltage levels, input/output (I/O) levels and an element maintenance bus (ETM-bus). These standards are intended to reduce the cost of using chips from different VHSIC contractors in one DoD system.

The standards developed for the module level are a system interconnect bus (PI-bus) and a system maintenance bus (TM-bus). Together, these standards provide the communications paths that will allow future DoD systems to be designed using open systems concepts in which modules from different contractors can be efficiently combined to configure new systems. These standards are already reducing DoD costs and development schedules by simplifying system development and promoting reuseability of modules across DoD systems. The PI-bus and TM-bus standards are compatible with VHSIC Phase 1 technology as well as VHSIC Phase 2 technology. As a result, these standards are being put to immediate use in DoD systems such as CVIS, LHX, ATA, and ATF. We recommend that DoD adopt the VHSIC bus standards presented herein as general DoD standards and continue applying these standards to DoD programs. This will facilitate future system upgrades to VHSIC Phase 2 technology.

Figure 1-1 illustrates the relationship between the interoperability standards and a typical DoD system. The PI-bus and TM-bus provide inter-module communications throughout a backplane. The PI-bus provides a communication path for system control messages and moderate bandwidth data. The multi-drop bus structure and supporting protocol form an efficient mechanism

for transferring short, low latency messages typical of system control. The PI-bus, although limited by transmission line propagation delays rather than BIU technology, still provides block data transfer rates in excess of 50 megabytes per second. The TM-bus interconnects Maintenance Controllers, such as IBM's DxMD, that are distributed throughout the system. The Maintenance Controllers, in turn, provide an interface to the local ETM-buses which extend the maintenance communication channel down to individual VHSIC chips. Together, the TM-bus and ETM-bus provide a cost effective, independent communications channel that supports error logging, diagnostics and system maintenance actions. Features of these buses and the rationale which guided their development are given in "2.0 PI-BUS STANDARD," "3.2 TM-bus" and "3.3 ETM-bus." The chip clocking, power supply, and input/output level standard is described in "4.0 ELECTRICAL AND PHYSICAL STANDARDS."

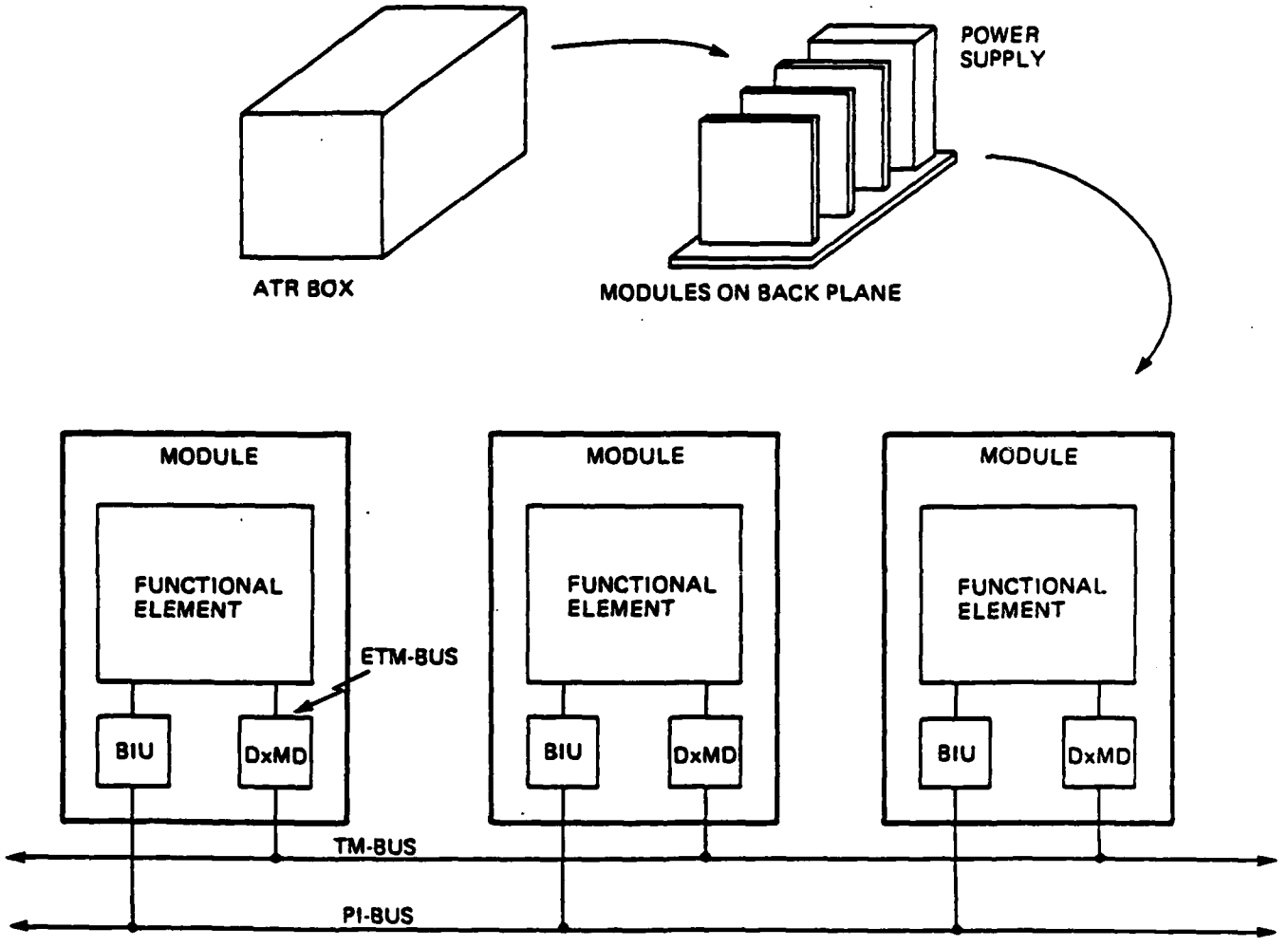


Figure 1-1. Interoperability Standards - System Environment

Section 2

PI-BUS STANDARD

2.1 INTRODUCTION

The PI-bus Specification has been developed to satisfy the requirement for a "standard interconnect system bus". Specifically, the PI-bus provides a multi-drop, module-to-module communications path for control messages and moderate bandwidth data.

The requirements for the PI-bus were based on the VHSIC Phase 2 brassboards, current DoD system designs and projected DoD system applications. Applications considered include flight control, image processing, general signal processing, spacecraft applications and general purpose distributed processing.

The preliminary PI-bus Specification was released to DoD for comment on May 10, 1985. An updated version of the specification is provided as Appendix B. PI-bus design rationale and the major features of the bus are summarized in the following section.

2.2 FEATURES

The PI-bus specification defines a linear, multi-drop, synchronous bus which supports digital message communications between up to 32 modules residing on a single backplane. Messages are transferred datum serial and bit parallel using a datum size of 16 bits (single word) or 32 bits (double word).

The PI-bus is specified at the physical and data link levels of the open systems interconnect reference model. By using the open systems interconnect model, we were able to develop a bus specification that is applicable to a large variety of DoD systems. Device functions are separated from PI-bus functions which are separated from the functions of higher levels of the open systems protocol (network layer and above). This approach lets us ensure that the PI-bus provides the necessary functions to the rest of the system without imposing unnecessary requirements on the rest of the system. In addition, this approach allows independent development of specifications for other system layers and isolates the VHSIC hardware from the many required variations of the higher level layers.

Figure 2-1 illustrates the conceptual model of the PI-bus and PI-bus modules. Each module attached to the PI-bus consists of a device which performs the application specific function of the module and a bus interface which implements the PI-bus master-slave communications protocol.

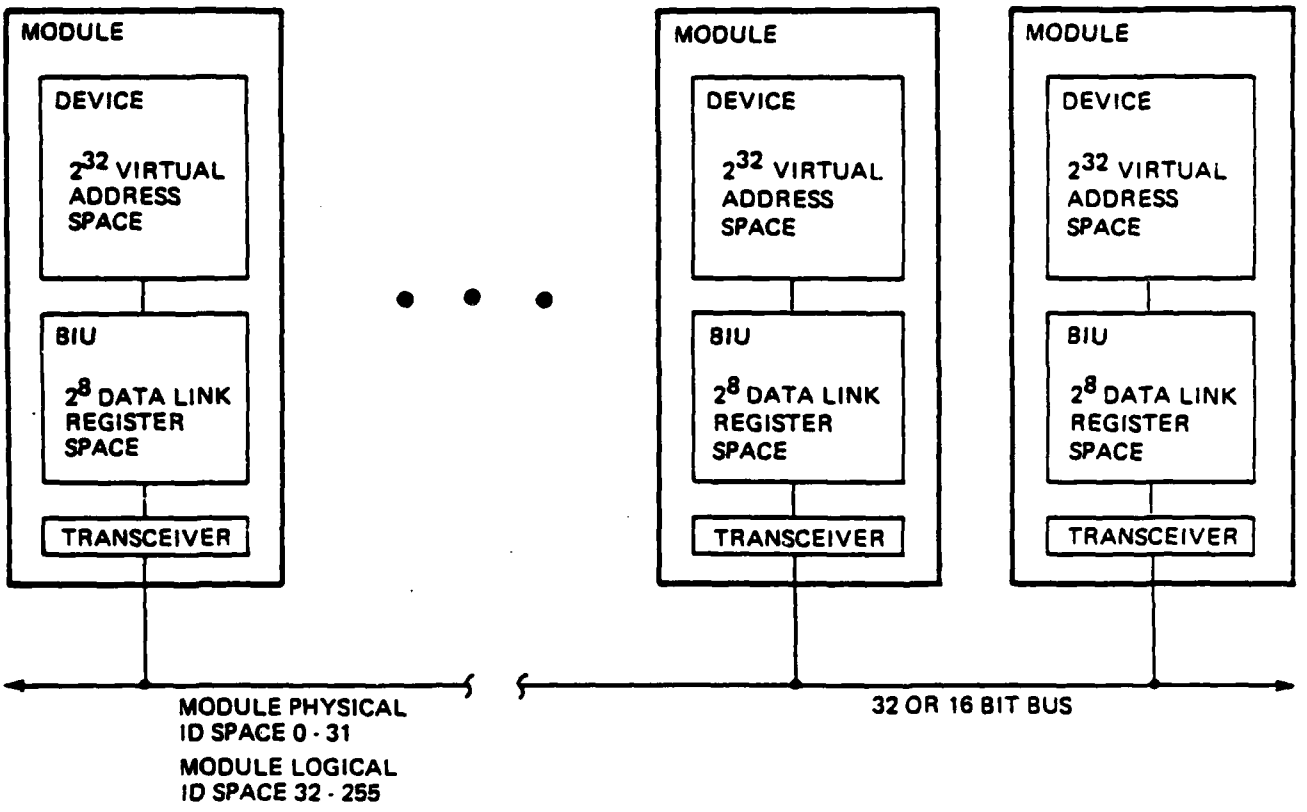


Figure 2-1. PI-bus Conceptual Model

The PI-bus provides separate address spaces for the device and bus interface segments of a module. This design avoids potential conflicts with the device's memory management system. The device itself is modeled as a virtual memory space with a 32 bit address range. Thus, future DoD systems with 32 bit address devices are fully supported. The bus interface is modeled as a separate memory space with 256 addressable data link registers. The control registers of the BIU are mapped into this address space. Six additional 256 word bus interface address spaces are provided to support user specific functions and higher level protocols. These address spaces have been made accessible from the bus via BIU command messages so that the configuration of the bus can be controlled at the system level.

PI-bus protocol allows the current bus master to communicate with one or more slave modules simultaneously. The slave modules are selected to participate in a message transfer by an 8-bit virtual module address called the slave identification (slave ID). Each module is assigned a unique physical identification code in the range of 0 to 31. This allows messages to be directed to specific modules and is particularly useful during system initialization and maintenance actions. Every module also recognizes identification code 32 as a broadcast indication. This slave addressing mode provides an efficient method for sending system wide messages. In addition, modules can recognize up to 223 logical slave identification codes. The system programmer's task can be simplified by assigning unique logical slave identification code(s) to each module on the basis of the tasks which that module is capable of performing. The programmer is then freed from concern about the module's physical address which may change with various system configurations or fault recovery actions. Since a logical slave identification can be assigned to more than one physical module, this address mode also provides a mechanism for multicast operations in which the master simultaneously writes to a selected subset of the modules on the bus. Thus the master can efficiently send messages to groups of modules.

As illustrated in Figure 2-2, the PI-bus protocol specifies a set of bus state transitions which control the communication sequences. The state sequences are designed to allow the bus to operate in a pipelined manner at the maximum clock rate allowed by the bus signal propagation delay. Master-slave handshaking is provided with a minimal performance penalty by operating the slave modules in synchronism with the master and using bus state look-ahead techniques. The slave modules are able to anticipate normal message events, such as acknowledge cycles, and respond on the same cycle that the master uses to signal the event. This provides a significant performance advantage over conventional buses which require several bus transit times for the slave to receive the master's signal, interpret the signal and send the appropriate response to the master. A number of independent messages may be transmitted during a bus master's tenure to minimize the overhead associated with acquiring bus mastership and ensure that the bus master can coordinate interdependent messages. For example, in a system which uses course grain data flow control techniques the master would issue produce and consume messages during a single bus tenure.

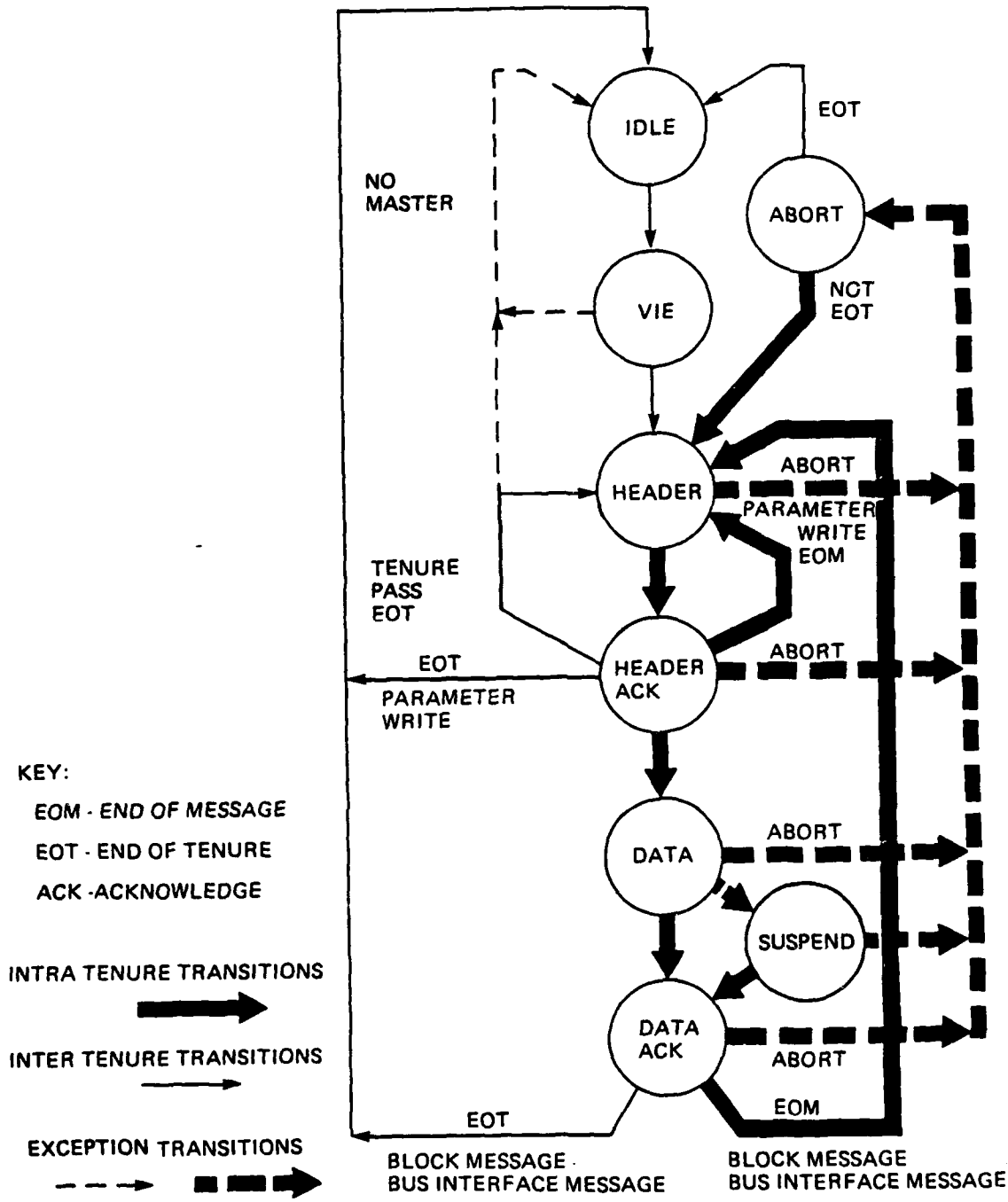


Figure 2-2. PI-bus Protocol State Diagram

A decentralized control structure was selected for the PI-bus to improve the reliability of the bus. No central module is required for operation of the bus and the PI-bus transceivers are designed to fail in a high impedance mode. Thus, the failure of one or more modules attached to the bus does not inhibit the operation of other modules on the bus. The arbitration mechanism allows potential bus masters to compete (vie) for mastership of the bus based on a 12 bit priority code. A large number of priority levels allows message priority to be resolved between individual tasks that may be concurrently executing on various modules rather than limiting priority resolution to a module level. A unique error detection and correction mechanism is incorporated into the arbitration procedure to maintain the integrity of the process. In addition to the Vie priority resolution sequence, the PI-bus provides a direct tenure pass mechanism. This allows the implementation of deterministic token passing systems such as those required for flight control.

A technique for temporarily suspending low priority block data transfers to reduce bus acquisition latency for higher priority messages is defined. This suspend/resume mechanism works in conjunction with a vie interval timer to allow system level control over bus acquisition latency. A bus master's tenure may be suspended to allow a higher priority message to be sent over the bus. After the higher priority message has been transferred, bus mastership can be returned to the original bus master and the suspended message transfer can be resumed. A current bus master can also suspend a low priority message to transmit a higher priority message. The number of levels of suspended messages is not limited by the bus protocol.

Extensive signal line and sequence error detection capability is incorporated into the bus definition. Five major classes of signal lines make up a PI-bus: 1) Data Group, 2) Cycle Type Group, 3) Acknowledge Set, 4) Wait and 5) Bus Request. Single line error correction is provided for each signal group. A compatible subset of the bus provides single line error detection and allows performance and line count to be traded against error correction capability to achieve the best overall system characteristics. The number of bus signals required to implement the full 32 bit bus with error correction is 58. A 16 bit bus with error correction requires 42 lines. For the error detection case, 16 and 32 bit buses require only 29 and 46 lines, respectively.

The PI-bus provides a mechanism which allows the master and slave(s) to control the transaction rate of the bus by inserting "wait" cycles. This mechanism allows the bus to support various types and performance levels of devices. A bus clock rate in excess of 12.5 MHz is feasible for a large bus configuration (32 modules and 24 inches). Higher clock rates are possible for shorter buses and/or fewer modules. For a 12.5 MHz bus clock, the PI-bus meets all defined performance benchmarks.

Section 3

SERIAL TEST AND MAINTENANCE BUS STANDARDS

3.1 INTRODUCTION

The serial test and maintenance bus has been defined on the basis of the VHSIC Phase 2 brassboards, current DoD systems designs and projected DoD system applications. The resulting specifications were prepared by IBM, Honeywell, and TRW with DoD direction to achieve interoperability goals and requirements.

The basic intent of the VHSIC Phase 2 Test and Maintenance bus (TM-bus) is to provide a standard communications path for transferring test and diagnostics messages and data between modules residing on a single backplane. The overall purpose is to standardize the test and maintenance interface so that a subsystem comprising multiple vendor chips and/or chip sets can be monitored by a single (general purpose) maintenance controller. To effectively address the chip-level communications path, the Element Test and Maintenance bus (ETM-bus) has been defined.

The TM-bus was designed to interface at the module level. An evaluation was made for three possible levels (i.e. the chip level, Multi-Chip Package (MCP) level, and the module level). After an evaluation of the pros and cons of each interface option, it was apparent that there were a number of problems with interfacing the overall TM-bus directly to every chip. Therefore, the TM-bus will connect to either Diagnostic Maintenance Devices (DxMD) or Test and Maintenance Processors (TMPs) or similar devices at the module level.

With DxMD/TMPs devices connected to the TM-bus, the DxMD/TMPs interface to individual vendor chips by means of a secondary bus known as the ETM-bus which is more appropriate at the chip level. This reduces the traffic/congestion on the TM-bus and increases the fault tolerance of the backplane TM-bus by reducing the number of lines and drops. In addition, hardware required on each VHSIC Phase 2 chip to communicate over the TM-bus is reduced while while test capabilities are enhanced.

With the TM-bus connected to a module via a DxMD type device, chip-level interoperability becomes more achievable with minimal impact to each contractor. Therefore, with the TM-bus connected at the module level, the following would hold true:

1. TM-bus transceivers are connected to a module via a DxMD or similar device (like Honeywell's TMPs).

2. Bus commands are easier to standardize at the module level.

3. Bus connections are reduced, which aids fault tolerance.

4. DxMDs are not required for each MCP.

5. Clock control and distribution would still be handled by the On-Chip

Monitor (OCM) which is controlled by the DxMD and the TM-bus MASTER.

6. Self-test of all of the chips would take less time since each DxMD can initiate testing on each module.

7. Interrupt handling becomes simpler since the DxMD screens data before notifying the TM-bus MASTER.

8. Chip-to-chip and MCP-to-MCP interconnect testing would be handled by the DxMD device while module-to-module interconnect testing would be controlled by the TM-bus MASTER.

9. The ETM-bus is more simple and efficient for chip test requirements without burdening the TM-bus.

In summary, the two level (TM-bus for module level and ETM-bus for chip level) bus structure has the following advantages:

1. Reduces on-chip interface circuitry.

2. Chip address identification is more straight forward.

3. Chip-to-chip interconnect testing is more efficient.

4. Allows support of vendor specific test technologies/designs.

5. Fault tolerant TM-bus variations are easier to implement.

6. Simplifies/standardizes DoD testing of various vendor modules.

7. Fewer connections on a single bus minimizes application problems.

    * Higher fault tolerance,
    * Less chip hardware for re-drive,
    * Does not limit maximum bus frequency, and
    * Speeds up interrupt screening/handling.

Some of the key test and maintenance bus features are summarized below.

1. Accommodates off-line and real-time testing.

    ** Interrupt capability for real-time applications.
    ** Provides for scan data (long data vectors) or test commands (short/simple) for off-line or background testing.

2. Protocols and standard commands kept simple.

    ** Hooks provided to accommodate contractor specific

requirements.

3. 4-wire system chosen for best compromise between implementation simplicity and system reliability.

4. Error detection (1 bit packet parity)

5. Provides interface capability with non-VHSIC Phase 2 modules.

The preliminary TM-bus Specification was released to DoD for comment on May 10, 1985. The preliminary ETM-bus Specification was released to DoD for comment on July 1, 1985. An updated version of the TM-bus Specification is provided as Appendix C and an updated version of the ETM-bus Specification is provided as Appendix D. Major features of both buses are summarized in the following sections.

3.2 TM-BUS

The TM-bus specification establishes the electrical, functional and performance requirements for the set of signal lines that constitute the Test and Maintenance Bus (TM-bus).

The purpose of this standard is to establish requirements for the TM-bus and facilitate interoperability of modules that use the TM-bus.

The TM-bus is intended as a serial path for test and maintenance control and data information.

The TM-bus is a linear, multi-drop communications medium which transfers bit serial data between a 'MASTER' module and up to 32 'SLAVE' modules residing on a single backplane.

TM-bus modules implement the TM-bus protocol and meets all requirements of the referenced specification.

Figure 3-1 illustrates the TM-bus and TM-bus modules. Conceptually, each module consists of a device that performs the application specific function of the module and a bus interface which implements the TM-bus master-slave communications protocol.

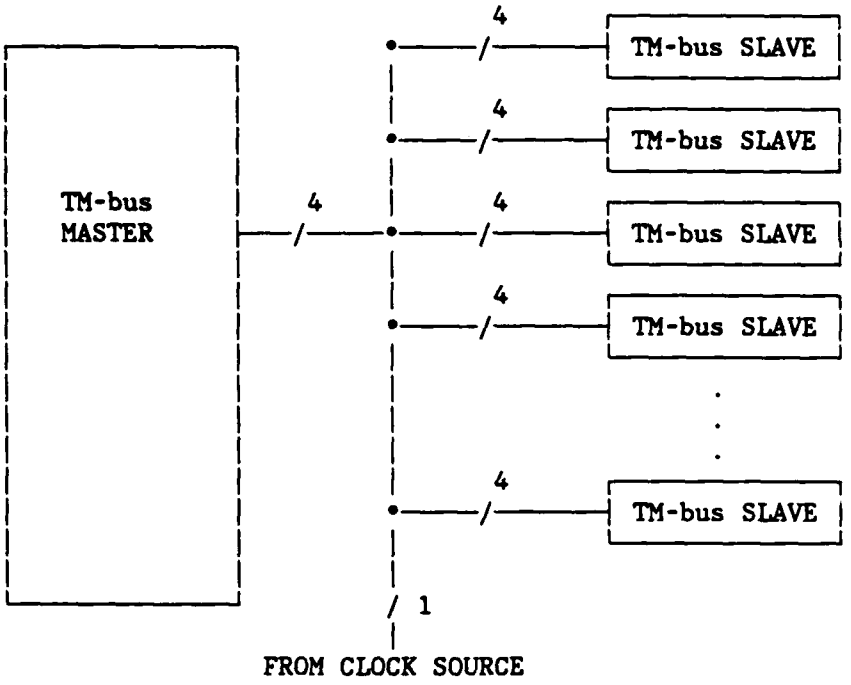


Figure 3-1. Conceptual Model Of Bus And Modules

The TM-bus signal, and clock lines are defined in the following paragraphs.

There are four signal types that make up the TM-bus as shown in Figure 3-2 on page 3-5. All bus signals shall use negative logic, i.e. the logic '1' state (or asserted state) is the lowest voltage level on the bus and the logic '0' (or released state) is the higher voltage level on the bus.

The TM-bus CLOCK signal is a single phase clock. The TM-bus interface shall support the full range of clock frequencies from zero (0) Hz to 6.25 MHz. All control and data transfer operations are synchronized with the TM-bus CLOCK signal. All data and commands are placed on the TM-bus on the high to low transition of the clock and latched-in on the next high to low transition.

The TM-bus MASTER DATA signal is a single uni-directional line used to transmit device addresses, instruction data, and/or scan data from the MASTER to the SLAVE(s). The MASTER DATA line is also used in conjunction with the CONTROL line to indicate bus states.

The TM-bus SLAVE DATA signal is used to transmit acknowledgements, data, and/or interrupts from the SLAVE(s) to the MASTER. The TM-bus SLAVE DATA line supports a wired-OR configuration.

The TM-bus CONTROL signal is a single uni-directional line from the MASTER to the SLAVES(s). When the CONTROL line is asserted the bus is placed in the DATA TRANSFER state. When CONTROL is released, the bus is in the PAUSE or IDLE state.

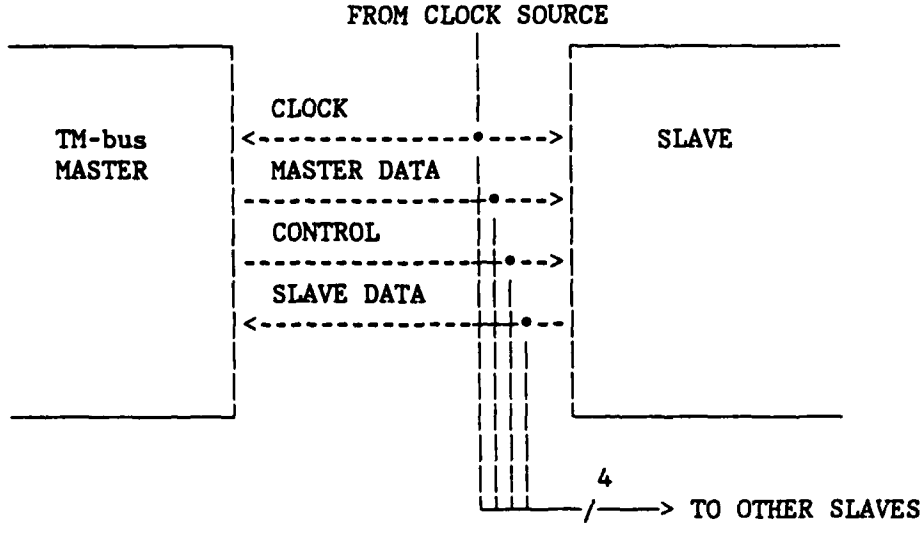


Figure 3-2. TM-bus Signals

Each TM-bus SLAVE is addressed by an eight bit address field. This address shall be sent in the HEADER packet containing the five (5) bit module address (bits <16..12>) and the three (3) bit sub-address (bits <11..9>).

The five-bit module address field in the HEADER shall be compared to five Module IDentification (MID) inputs to determine if the SLAVE is being addressed. As a minimum, each SLAVE shall also have a Module Identification Parity (MIP) input that shall be set such that the modulo two sum of the five MID inputs and the MIP input equals one (1). (Note: The asserted state of each input is a logical one.) When used in conjunction with the VHSIC Phase 2 PI-Bus, it is recommended that each TM-bus SLAVE module shall have its MID and MIP inputs hardwired to the backplane of the TM-bus (reference Section "4.2.4 Module Identification" of the PI-bus Specification, Version 2.1 dated September 25, 1986). If an unrecoverable error occurs on the MID inputs, the TM-Bus SLAVE shall not execute any commands and shall release

the SLAVE DATA line.

Comparison of the three (3) sub-address bits from the HEADER packet against Sub-address IDentification (SID) inputs is optional.

Module addresses '0' through '30' have a maximum of eight subaddresses. Address '31' is limited to three subaddresses ('F8', 'F9', and 'FA' HEX) due to restrictions of broadcast and multicast commands.

The Test and Maintenance Bus (TM-bus) is the channel for control and data information flow between the TM-Bus MASTER (e.g., a maintenance controller) and SLAVE modules within a system. The module in control of the TM-bus is referred to as the MASTER and all other modules on the TM-bus are referred to as SLAVES. The information transferred and the scheduling of data and commands are system dependent and is not addressed in the specification. Figure 3-3 on page 3-7 summarizes the TM-bus design parameters and characteristics.

o Performance Characteristics	o Protocol Characteristics
- 6.25 MHz clock (Typical)	- 8 reserved address bits
- 4 pin bus signals	- 32 module addresses (maximum)
- Synchronous Operation	- 8 sub-addresses per module address
- Two Data Lines	- Multi-drop Configuration
- SLAVE status register	- Interrupt Capability

Figure 3-3. TM-bus Design Parameters and Characteristics

For additional details on the bus protocol, standard commands, error/status reporting, etc. please refer to the TM-bus specification.

3.3 ETM-BUS

The ETM-bus specification establishes the electrical, functional and performance requirements for the set of signal lines that constitute the

Element Test and Maintenance Bus (ETM-bus).

The purpose of the standard is to establish requirements for the ETM-bus and facilitate interoperability of VLSI chips which use the ETM-bus.

The ETM-bus is intended as a serial path for test and maintenance control and data information at the chip level.

The ETM-bus is a communications media which transfers bit serial data between a 'MASTER' device (CONTROLLER) and up to 32 logical 'SLAVE' elements interfacing to a single ETM-bus CONTROLLER.

Figure 3-4, illustrates the ETM-bus and ETM-bus elements. Conceptually, each element consists of a device which performs the application specific function of the chip and a bus interface which implements the ETM-bus master-slave communications protocol.

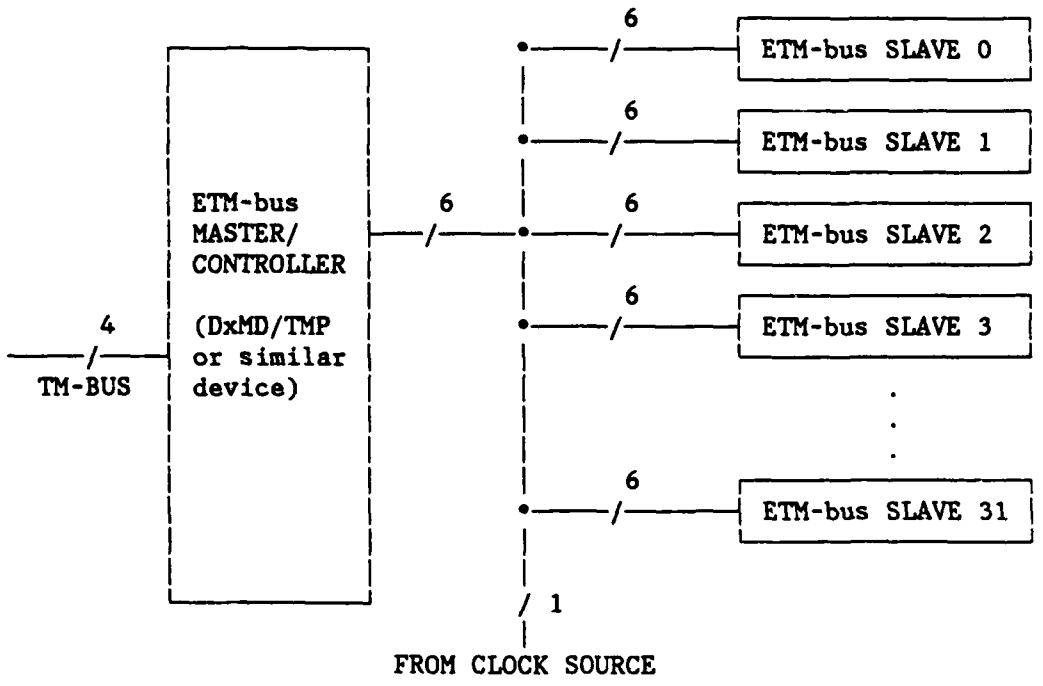


Figure 3-4. Conceptual Model Of ETM-bus Elements

The ETM-bus signal and clock lines are defined in the following paragraphs.

Lines are designated by name. When a set of related bits are represented by the same name, the bits within the set are differentiated by number with the most significant bit (MSB) numbered 0. All fields are referred to by their bit position within a data word transferred over the ETM-bus. All ETM-bus signals are defined in the following paragraphs. A "-" symbol associated with a signal means that the signal is active low.

There is a minimum of six signal types that make up the ETM-bus as shown in Figure 3-5 on page 3-9. Additional lines, to accommodate enhanced application requirements, are not precluded. SELECT and INTERRUPT are negative logic and all other signals shall use positive logic.

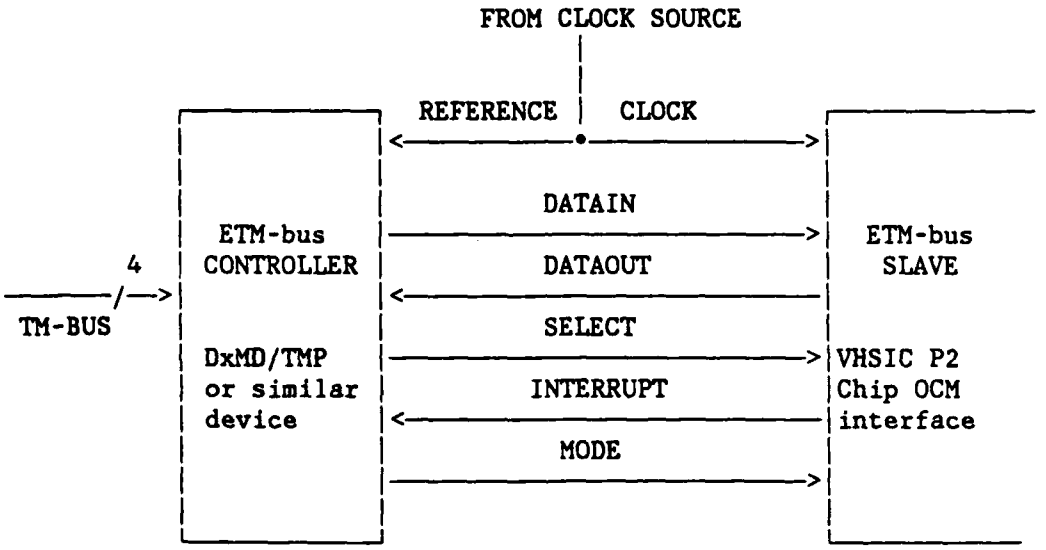


Figure 3-5. ETM-bus Signal Types

All data transfer operations are synchronous with the REFERENCE CLOCK (REF CLK) signal. All bus activity shall be relative to the high-to-low transition of the REFERENCE CLOCK. The CLOCK signal will be single phase. The ETM-Bus interface should support the full range of clock frequencies from zero (0) to 6.25 MHz.

The ETM-bus DATAIN signal is a single unidirectional line into the SLAVE. Instruction data and scan data are transmitted to the SLAVEs over the DATAIN line. In the star configuration, the DATAIN signal is sourced from the CONTROLLER. In the ring configuration, the DATAIN signal is sourced from either the CONTROLLER or another SLAVE.

The ETM-bus DATAOUT signal is a single unidirectional line from a SLAVE. In the star configuration, the DATAOUT signal is transmitted from the SLAVE to the CONTROLLER. In the ring configuration, the DATAOUT signal is transmitted from the SLAVE to either the CONTROLLER or the DATAIN pin of another SLAVE. The DATAOUT line supports three-state operation. The DATAOUT signal will be in the high-impedance state when inactive (e.g., not in a logic 1 or 0 state).

The SELECT signal line (-SEL) is unidirectional from the CONTROLLER to the SLAVEs. The SELECT signal line defines when data transfer operations are to occur. SELECT becomes asserted (low) one cycle before instruction or scan data is serially transferred across the DATA lines. SELECT is released one cycle before the end of a data transfer. In a ring bus structure, where all SLAVEs share a common SELECT, all SLAVEs are selected simultaneously. In a star bus structure, all SLAVEs have a separate SELECT line.

INTERRUPT (-INT) is unidirectional from the SLAVES to the CONTROLLER. The INTERRUPT line is asserted (low) to indicate that an event (such as an error or other predetermined condition) has occurred. The INTERRUPT line remains asserted until the interrupt is serviced. The INTERRUPT line shall support a wired-OR configuration. The INTERRUPT line may operate asynchronously.

The ETM-bus MODE signal is unidirectional from the CONTROLLER to the SLAVE. MODE shall be used to establish the type of operation that is performed when SELECT is being asserted (see Figure 3-6). MODE must be valid one cycle before instruction or scan data is serially transferred across the DATA lines and remain stable for the length of the transfer. The MODE line shall stay valid as long as SELECT is asserted.

MODE	OPERATION
0	INSTRUCTION/STATUS
1	SCAN

Figure 3-6. ETM-bus MODE Line Definition

The use of the ETM-bus shall be defined by the level of the SELECT line and the MODE line. Two types of data transfer operations can take place on the ETM-bus as specified in Figure 3-6 on page 3-10. Data transfers are to

end by no more than one (1) clock cycle after the SELECT line is released (high).

The Element Test and Maintenance Bus (ETM-bus) is a channel for control and data information flow between a module maintenance controller and the individual elements (e.g., VHSIC Phase 2 chips) within the module. The module maintenance controller may be implemented as a Test and Maintenance Bus (TM-Bus) SLAVE as described in the TM-Bus specification. The module maintenance controller is referred to as the CONTROLLER and all other elements on the ETM-bus are referred to as SLAVEs. The information transferred and the scheduling of data and instructions is system dependent and is not addressed in the specification. Figure 3-7 on page 3-11 summarizes the ETM-bus design parameters and characteristics.

o Performance Characteristics	o Protocol Characteristics
- 6.25 MHz clock (Typical)	- Each ETM-bus logically supports up to 32 SLAVE devices
- Unidirectional data lines	
- Minimum of 6 pin bus signals	- Supports ring and star configurations
- TTL compatible	- Interrupt Capability

Figure 3-7. ETM-bus Design Parameters and Characteristics

For additional details on the bus protocol, commands, error/status reporting, etc. please refer to the ETM-bus specification.

Section 4

ELECTRICAL AND PHYSICAL STANDARDS

4.1 INTRODUCTION

The basic electrical and physical requirements for supporting the PI-bus, the TM-bus and the ETM-bus are contained in the specifications for those buses. The rationale for those requirements is discussed below. The status of work on the chip power supply and Input/Output (I/O) voltage level standards is given in "4.3 Chip Power Supply And I/O Voltage Standards."

4.2 PI-BUS AND TM-BUS ELECTRICAL DESIGN

Design considerations and rationale for the electrical design of the PI-bus and TM-bus are given below. Representative results of simulations that were performed to support the electrical design are provided in Appendix A.

High performance multi-drop buses represent a significant electrical design challenge. A bus that is capable of interconnecting modules across a backplane at transaction rates in excess of a few megahertz must operate in a transmission line environment. The electrical design must focus on the issues of bus driver power, signal reflections, noise and signal propagation time on the bus. Since the power required to drive the bus precludes direct drive from the VHSIC BIU chip, these issues are independent of the use of VHSIC Phase 2 technology. Thus, the minimum transaction rate of the bus is primarily determined by the characteristics of the backplane transmission line and the bus transceiver.

The distributed inductance and capacitance of the signal lines in the backplane determine the characteristic impedance of the transmission line and the propagation delay of signals on the line. The characteristic impedance (Z0) of the line is given by:

$$Z0 = SQRT(L/C)$$

and the propagation delay of the line per unit length is

$$T0 = SQRT(L*C)$$

where L is the inductance and C is the capacitance of the line per unit length.

A high characteristic impedance is desirable to minimize the power required to drive the transmission line. To minimize the propagation delay, the inductance and capacitance of the line must be minimized. In practice, these quantities are limited by the dielectric material and line spacing used in the backplane. Typical values for an unloaded transmission line are

L = 8.6 nanohenries per inch and
C = 2.9 picofarads per inch

which yield

Z0 = 54.5 ohms and
T0 = 0.16 nanoseconds per inch or 1.9 nanoseconds per foot.

As shown in Appendix A, two point nets using this type of line can be driven

by standard TTL devices such as the 54F244 transceiver.

The situation is quite different for signal lines required on a multi-drop bus. The distributed capacitance added to the transmission line when modules are plugged into the backplane drastically lowers the characteristic impedance of the line and increases the propagation delay for signals on the line. For a conventional bus design based on TTL or CMOS drivers, each module adds approximately 30 picofarads to the line capacitance. For a typical module spacing of 0.65 inches, this represents an additional load of 46 picofarads per inch. The loaded transmission line characteristic impedance, Z, becomes approximately 13 ohms and the loaded propagation delay, T, becomes 0.65 nanoseconds per inch (7.8 nanoseconds per foot). Furthermore, the modules no longer drive the bus line from the end of the net. As a result, the modules must drive one-half the characteristic impedance of the line. For our example, this is only 13/2 = 6.5 ohms. This low impedance precludes initial wave switching with conventional drivers. Without initial wave switching, several round trip delays on the bus (at 31 nanoseconds each for a two foot long bus) may be required for the signal to settle. The settling time is extended by the lack of proper bus termination. However, if termination close to the characteristic impedance of the bus is used, TTL drivers are not capable of maintaining acceptable DC voltage levels. If the size of the driver transistor is increased to provide initial wave switching and acceptable DC levels, their output capacitance increases. This lowers the effective impedance of the transmission line, requiring still higher drive currents. Thus, another design approach is required to obtain incident wave switching.

For VHSIC, we have adopted an electrical design derived from the IEEE Futurebus (P896). The output capacitance of the bus drivers is reduced by using open collector drivers and isolating the drive transistor capacitance from the bus with a series Schottky diode. This produces a nominal low voltage on the bus of one Volt. To minimize power consumption, a termination voltage of two Volts was selected. The one Volt logic swing improves performance and reduces the power required to drive the bus as compared to a typical TTL design which has a logic swing of three Volts or CMOS which has a logic swing from ground to the power supply rail (typically 3 to 5 Volts). Noise margins are maintained by using a differential receiver circuit which has a narrower threshold range than conventional TTL receivers and by setting the receiver threshold in the center of the logic swing. In addition, the receiver provides noise rejection for pulses up to 4 nanoseconds wide. Using this transceiver design, the load which a module places on the bus is significantly reduced. For a typical module capacitance of 12 picofarads and a 0.65 inch module spacing, Z becomes approximately 20 ohms and T is 0.43 nanoseconds per inch (5.2 nanoseconds per foot). With this improved design, initial wave switching is achieved.

The termination impedance is typically chosen to be slightly higher than the characteristic impedance of the line to further reduce power consumption. For a 20 ohm line, a termination impedance of 30 ohms at each end of the bus is sufficient to control reflections on the bus and provide a low-to-high transition time that matches the high-to-low transition time. The PI-bus and TM-bus specifications allow termination impedances between 30 and 40 ohms to permit power/performance optimization for various bus config-

urations.

The bus simulations described in Appendix A show that for a single active driver, the settling time for a VHSIC bus design is approximately 13 nanoseconds versus approximately 41 nanoseconds for a conventional TTL design. For multi-drop buses, there are multiple active drivers that may produce a "wire-ored glitch" and extend the bus settling time. The glitch may occur whenever one module releases a bus line and another module drives the line. Thus the "wire-ored glitch" occurs on both wire-ored and three-state buses. The VHSIC bus design minimizes "wire-ored glitch" effects. However, the settling time may still be extended by up to the transit time of the bus (T times the bus length). For a typical bus length of 20 inches, this adds approximately nine nanoseconds to the bus delay and produces a total bus delay of 22 nanoseconds.

The power required to achieve this performance on a multi-drop backplane bus is excessive for a single VHSIC chip. Therefore, separate bus transceivers will generally be used to drive the bus. By decreasing the size and power consumption of the VHSIC BIU, the external transceivers will decrease overall cost and increase the reliability of the BIU chip.

We have conceived a bus transceiver that provides additional advantages over current designs by incorporating a transparent latch between the BIU and the bus driver device. The transceiver configuration and timing are illustrated in Figure 4-1. The transparent latch allows the BIU to drive the wired-or bus lines during the first half of the bus cycle and to independently sense the actual bus signal value during the second half of the bus cycle. As a result, the BIU does not require special timing signals or extra bus data input pins. Also, failures in the interface circuit can be detected by the associated BIU.

Signetics has developed an octal PI-bus transceiver (PN 54F776) which provides the transparent latch.

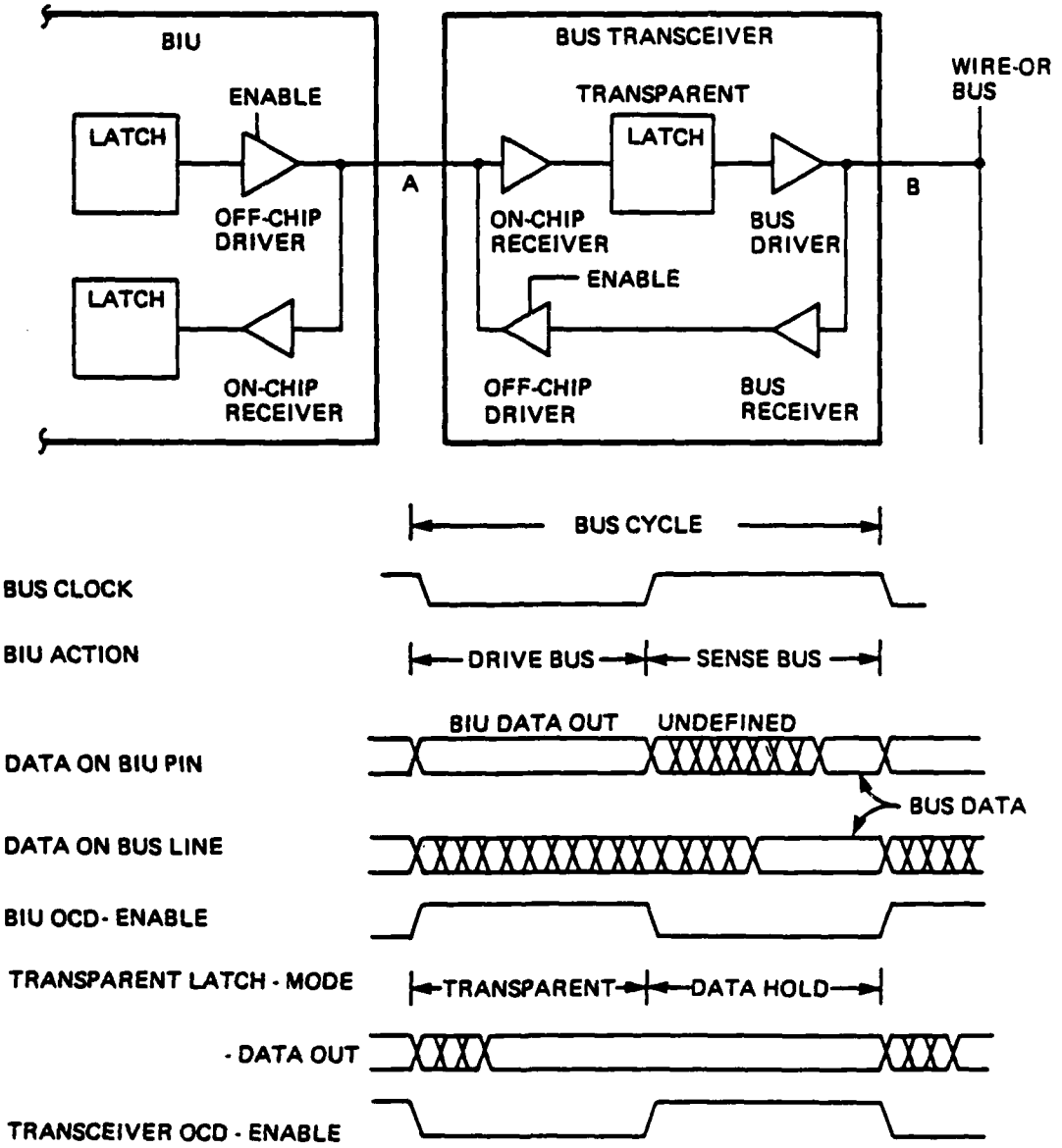


Figure 4-1. Bus Transceiver And Timing Diagram

4.3 CHIP POWER SUPPLY AND I/O VOLTAGE STANDARDS

VHSIC Phase 2 interoperability standards have been established for chip power supply, I/O voltage and clock requirements. These standards are contained in Appendix E.

4.3.1 ETM-BUS ELECTRICAL DESIGN

The ETM-bus is expected to be driven directly from VHSIC chips using the VHSIC chip I/O standard.

Appendix A

BUS ELECTRICAL SIMULATION

A.1 INTRODUCTION

A number of bus configurations have been simulated using the Advanced Statistical Analysis Program (ASTAP - a standard IBM program product). The simulations included conventional TTL driven buses and two point nets for clock distribution in addition to the open collector terminated bus finally selected for VHSIC use. Representative results are given below.

A.2 TRANSCEIVER CHARACTERISTICS

To establish a reference design, a conventional unterminated bus using a standard 54F244 transceiver was simulated using actual device characteristics. The open collector transceiver design selected for VHSIC was then simulated using device characteristics based on the 54F244. Figure A-1 illustrates measured output current versus output voltage for the low-to-high transition of the three-state 54F244 transceivers. Figure A-2 illustrates measured output current versus output voltage for the high-to-low transition and Figure A-3 represents the input current versus input voltage characteristic of the 54F244 transceivers. The characteristic curve labeled CHAR-A represents the worst case device operation at low temperature. This characteristic was used for simulation of the tri-state unterminated bus and an unterminated, single load net representative of the bus clock distribution path. The characteristic curve labeled CHAR-B was used to represent the pull-down characteristic for the output transistor in the open collector bus transceiver selected for VHSIC use. Approximate voltage versus current characteristics for the Schottky diode were combined with CHAR-B to produce the final open collector bus transceiver characteristics used in the simulations. The improved drive current represented by CHAR-B was selected to achieve incident wave switching for the VHSIC PI-bus and TM-bus with good noise margins. To achieve this current capability, the VHSIC bus transceiver will use a slightly larger transistor than does the 54F244 transceiver.

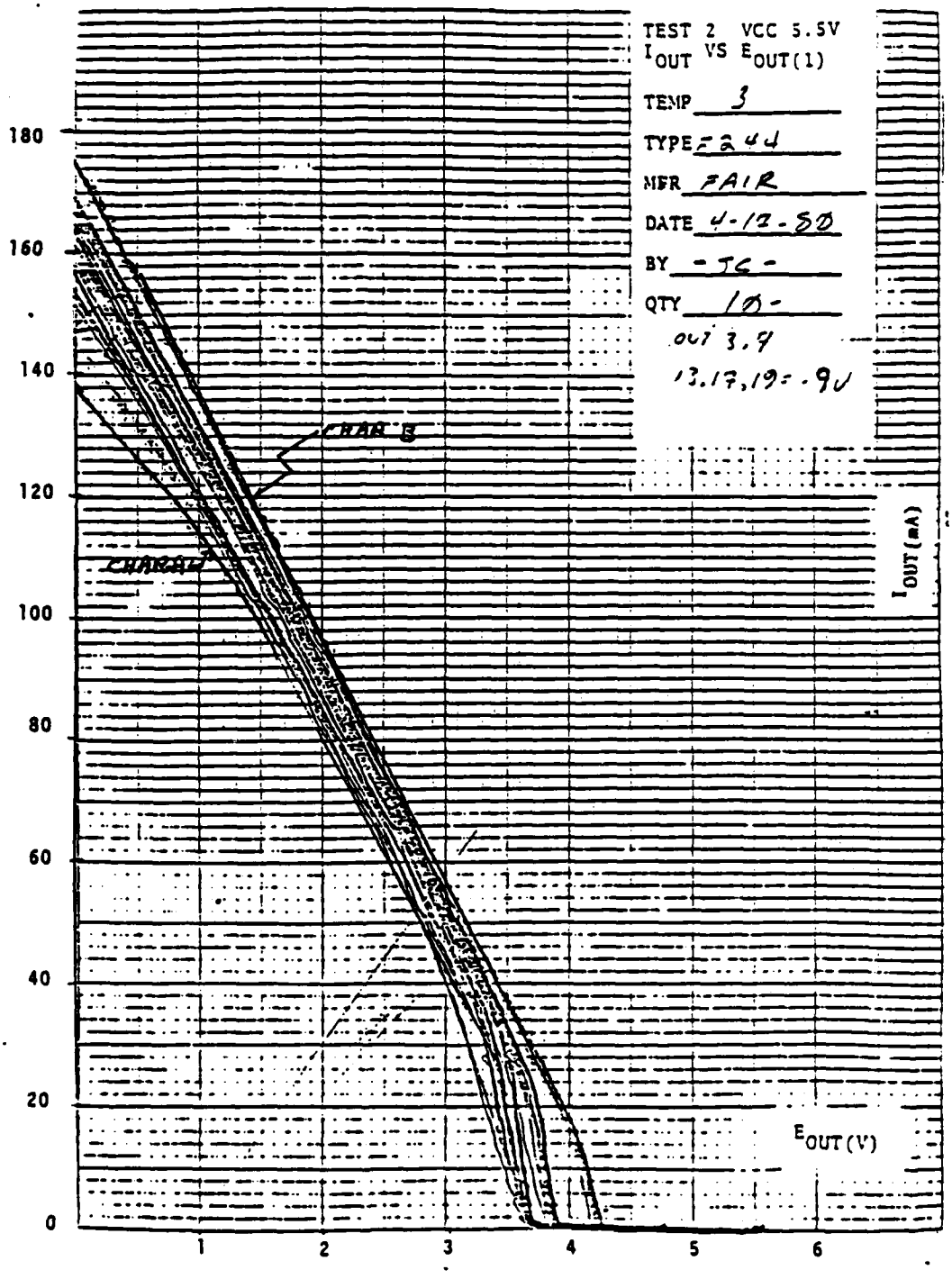


Figure A-1. Driver Pull-Up Current Versus Voltage

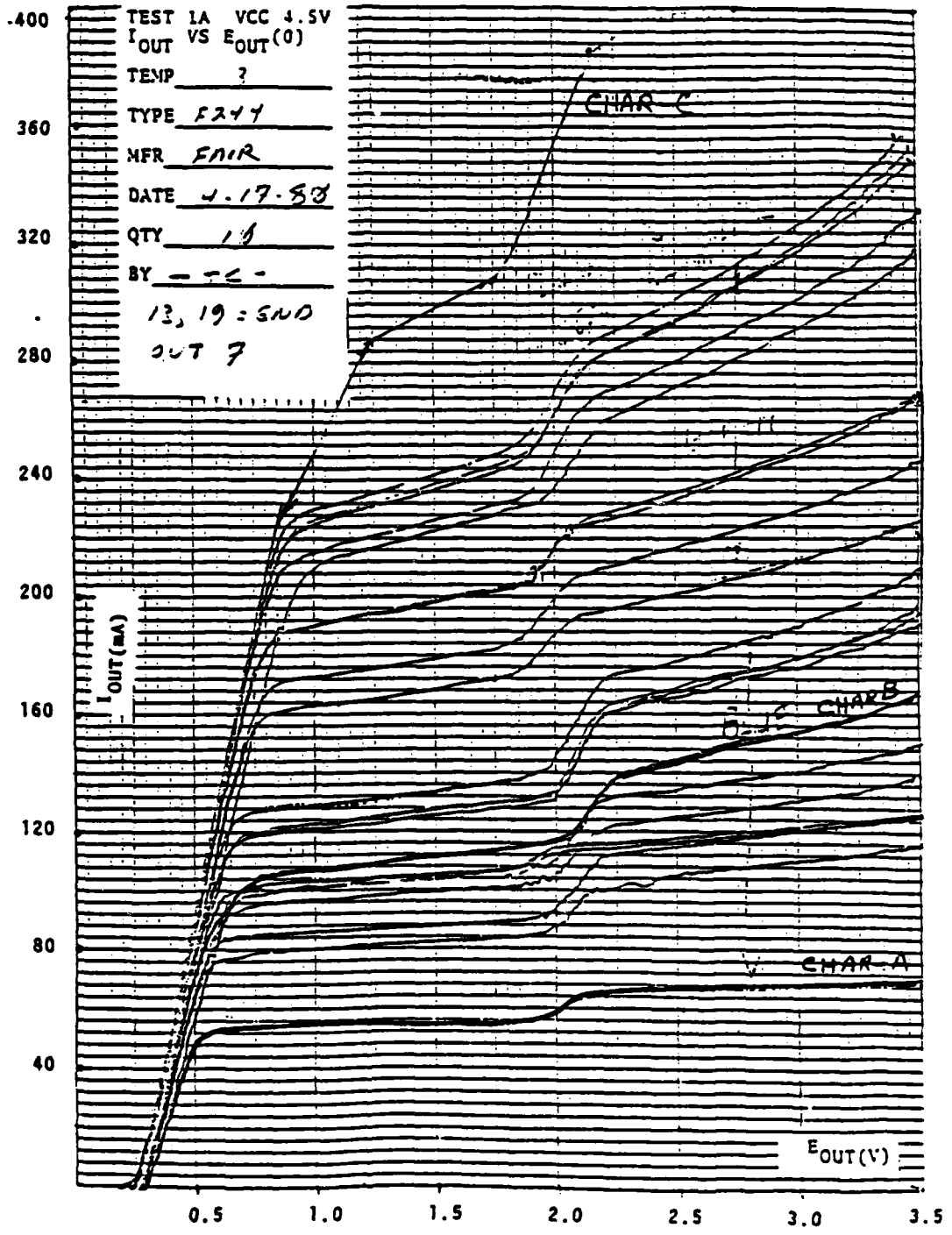


Figure A-2. Driver Pull-Down Current Versus Voltage

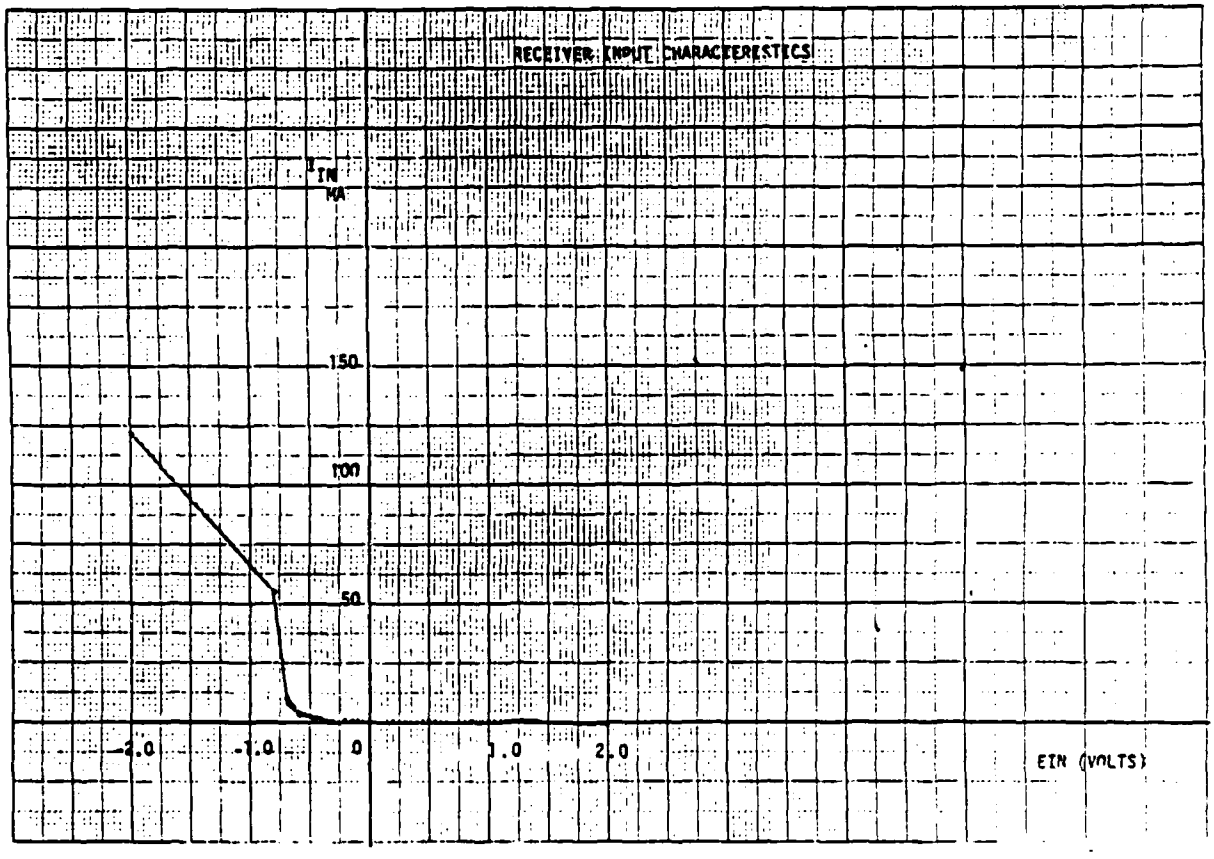
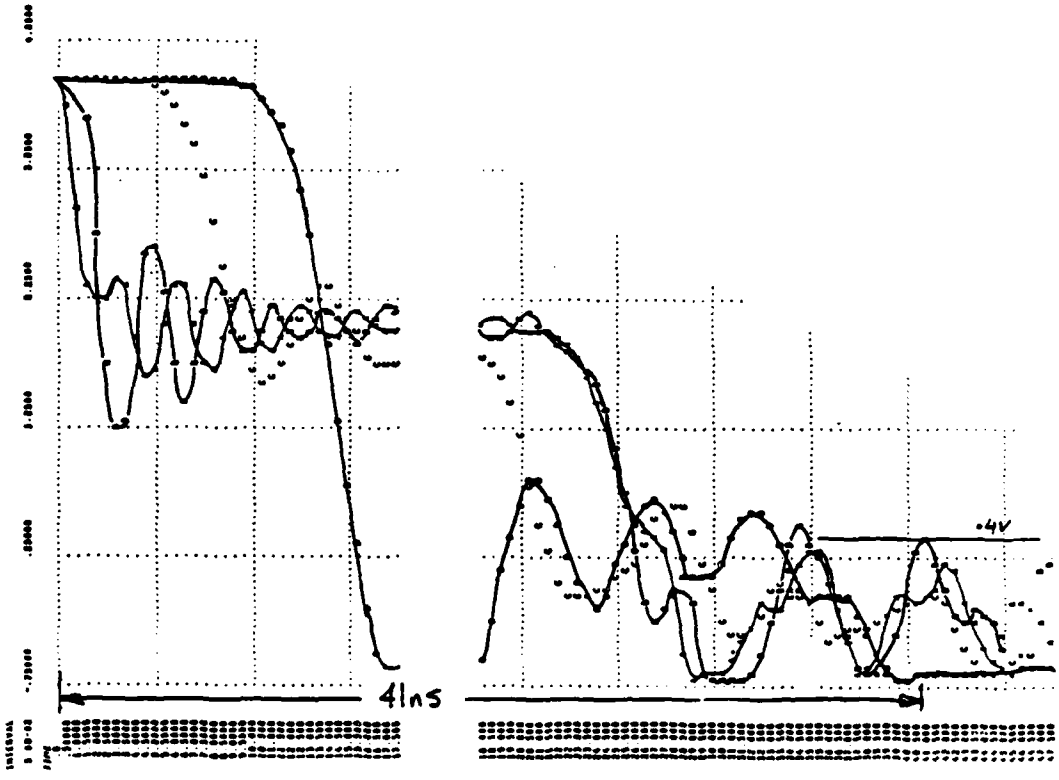


Figure A-3. Receiver Input Current Versus Voltage

A.3 SIMULATION RESULTS AND ASTAP MODEL

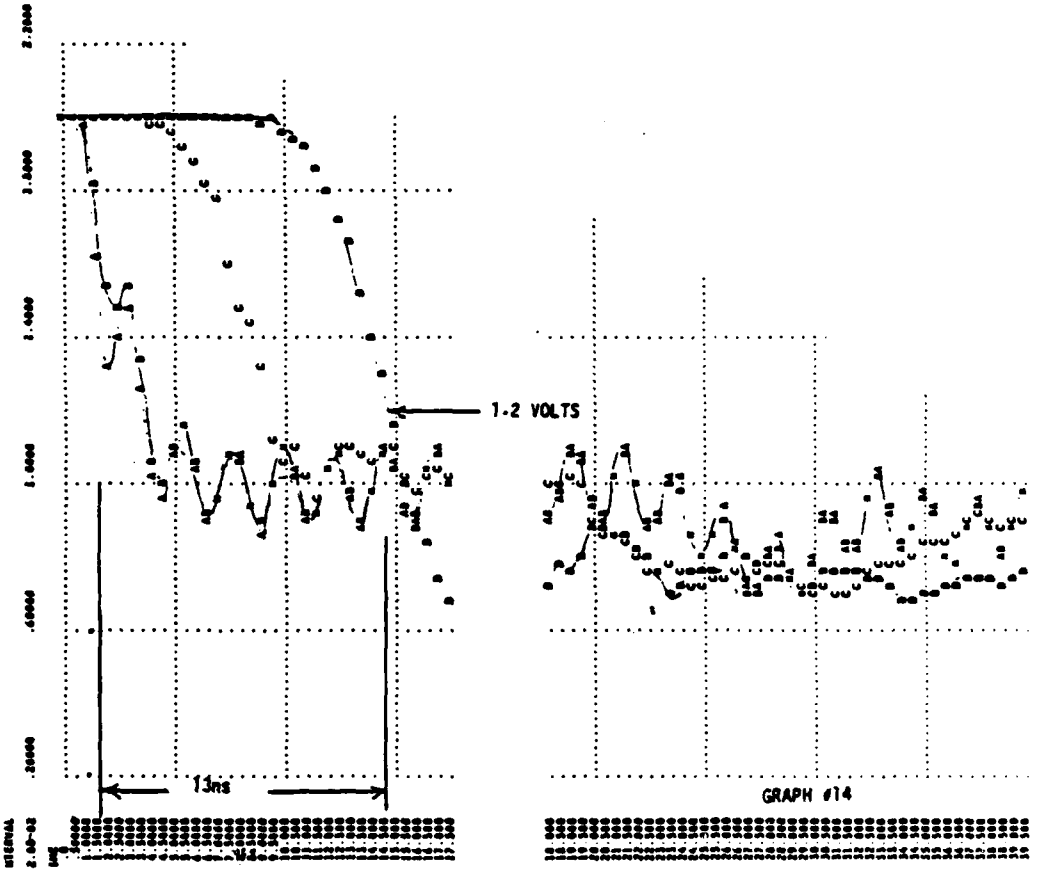
Figure A-4 illustrates the high-to-low transition for a 32 drop unterminated line driven by a 54F244 transceiver. The driver is not capable of switching the line below the receiver threshold of 0.4 Volts on the initial wave and there are multiple reflections. As a result the settling time is 41 nanoseconds. The low-to-high transition (not shown) is somewhat faster at 21.5 nanoseconds.

Figure A-5 illustrates the high-to-low transition for the open collector bus transceiver selected for VHSIC. The signal line is 20 inches long, has a characteristic impedance of 20 ohms, and is terminated through 30 ohms to +2 Volts at each end of the bus. The VHSIC transceiver drives the line below the receiver's low input threshold of 1.45 Volts on the initial wave to achieve a settling time of 13 nanoseconds. The low-to-high transition for this bus configuration is illustrated in Figure A-6. In this case the settling time is only 12 nanoseconds. The ASTAP model used for these simulations is given in Figure A-7.



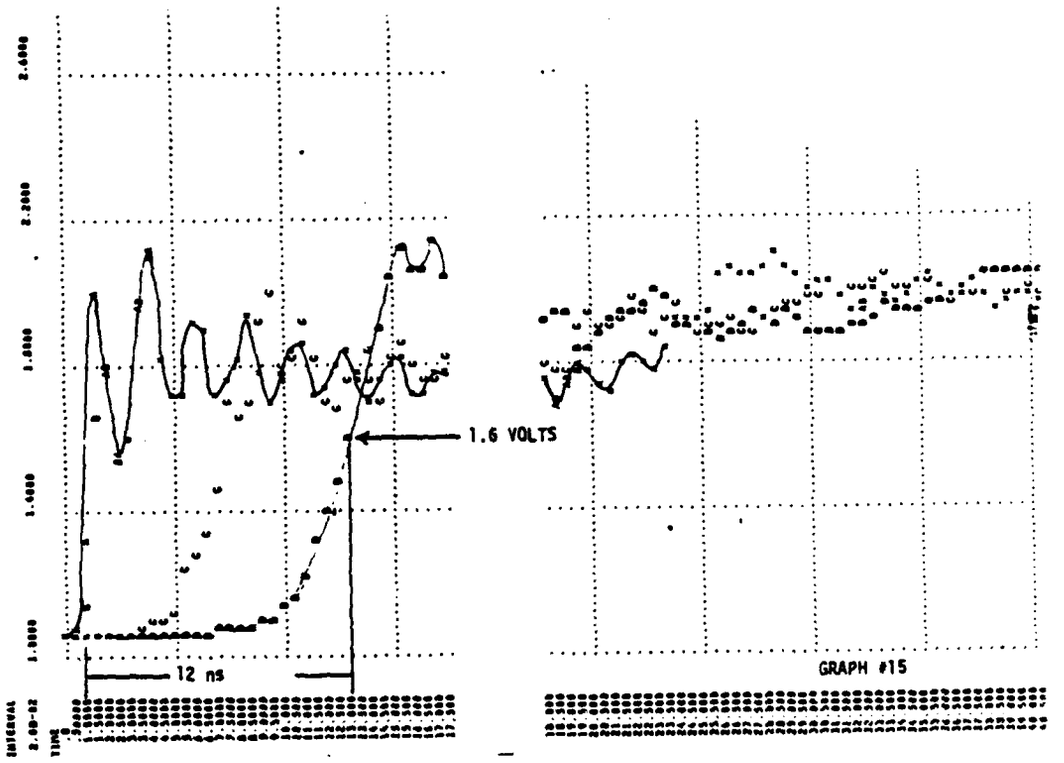
UNTERMINATED LINE
TTL 54F244 TRANSCEIVER
32 Drops (Nodes)
A - Near End Driver Node 1
B - Receiver Node 2
C - Receiver Node 17
D - Far End Receiver Node 32
5 Nanoseconds Per Division

Figure A-4. Unterminated 54F244 Driven Bus (High-to-Low)



TERMINATED LINE
OPEN COLLECTOR TRANSCEIVER
32 Drops (Nodes)
A - Near End Driver Node 1
B - Receiver Node 2
C - Receiver Node 16
D - Far End Receiver Node 32
5 Nanoseconds Per Division

Figure A-5. Terminated Open Collector Driven Bus (High-to-Low)



TERMINATED LINE
OPEN COLLECTOR TRANSCEIVER
32 Drops (Nodes)
A - Near End Driver Node 1
B - Receiver Node 2
C - Receiver Node 16
D - Far End Receiver Node 32
5 Nanoseconds Per Division

Figure A-6. Terminated Open Collector Driven Bus (Low-to-High)

```
                                L I S T   O F   D A T A   C A R D S
                                ###################################

        COLUMN 0          1         2         3         4         5         6         7          8
               1234567890123456789012345678901234567890123456789012345678901234567890123456789012  34567890
CARD NO                                                                                            COMMENTS

        ####0##########1##########2##########3##########4##########5##########6##########7##    ########8######
        PART 1 OF THE INPUT DATA :

     1           MODEL DESCRIPTION

     2 C
        ----0---------1---------2---------3---------4---------5---------6---------7--      -------8-----
     3           MODEL GETC1 ( )

     4 C         IBM INTERNAL USE ONLY

     5           ELEMENTS
     6            R0,Y-Z=0.055
     7            EDD2,GND-52=2.
     8            RT2,52-32= .83
     9            D1S = MODEL S54F244B(X-GND) (PD=1)
    10            L1S = MODEL S54FIN (X-GND)
    11            D1I = MODEL S54F244B(Y-GND) (PD=0)
    12            EB,GND-Z =((VRIC.D1)+(R0*PJDX.D1S))
    13            JS,W-GND = (PJDX.D1S)
    14            RS,W-GND = 1
    15            PJS = (JS)
    16            LL1 = MODEL MDIODE (1-66)
    17            D1 = MODEL M54F244B(66-GND-X-Y) (A=1, PD=0,PR0=0.055)
    18            L1 = MODEL M54FIN (2 -GND)
    19            L2 = MODEL M54FIN (3 -GND)
    20            L3 = MODEL M54FIN (4 -GND)
    21            L4 = MODEL M54FIN (5 -GND)
    22            L5 = MODEL M54FIN (6 -GND )
    23            L6 = MODEL M54FIN (7 -GND )
    24            L7 = MODEL M54FIN (8 -GND )
    25            L8 = MODEL M54FIN (9 -GND )
    26            L9 = MODEL M54FIN (10-GND )
    27            L10= MODEL M54FIN (11-GND )
    28            L11= MODEL M54FIN (12-GND )
    29            L12= MODEL M54FIN (13-GND)
    30            L13= MODEL M54FIN (14-GND)
    31            L14= MODEL M54FIN (15-GND)
```

Figure A-7. ASTAP Model For Open Collector Driven Bus

```
32          L15= MODEL M54FIN (16-GND )
33          L16= MODEL M54FIN (17-GND )
34          L17= MODEL M54FIN (18-GND )
35          L18= MODEL M54FIN (19-GND )
36          L19= MODEL M54FIN (20-GND )
37          L20= MODEL M54FIN (21-GND )
38          L21= MODEL M54FIN (22-GND )
39          L22= MODEL M54FIN (23-GND )
40          L23= MODEL M54FIN (24-GND )
41          L24= MODEL M54FIN (25-GND )
42          L25= MODEL M54FIN (26-GND )
43          L26= MODEL M54FIN (27-GND )
44          L27= MODEL M54FIN (28-GND )
45          L28= MODEL M54FIN (29-GND )
46          L29= MODEL M54FIN (30-GND )
47          L30= MODEL M54FIN (31-GND )
48          L31= MODEL M54FIN (32-GND )
49          TX1 = RLINE G52 (1-GND-2) (PL=0.65)
50          TX2 = RLINE G52 (2-GND-3) (PL=0.65)
51          TX3 = RLINE G52 (3-GND-4) (PL=0.65)
52          TX4 = RLINE G52 (4-GND-5) (PL=0.65)
53          TX5 = RLINE G52 (5-GND-6) (PL=0.65)
54          TX6 = RLINE G52 (6-GND-7) (PL=0.65)
55          TX7 = RLINE G52 (7-GND-8) (PL=0.65)
56          TX8 = RLINE G52 (8-GND-9) (PL=0.65)
57          TX9 = RLINE G52 (9-GND-10)(PL=0.65)
58          TX10= RLINE G52(10-GND-11)(PL=0.65)
59          TX11= RLINE G52(11-GND-12)(PL=0.65)
60          TX12= RLINE G52(12-GND-13)(PL=0.65)
61          TX13= RLINE G52(13-GND-14)(PL=0.65)
62          TX14= RLINE G52(14-GND-15)(PL=0.65)
63          TX15= RLINE G52(15-GND-16)(PL=0.65)
64          TX16= RLINE G52(16-GND-17)(PL=0.65)
65          TX17= RLINE G52(17-GND-18)(PL=0.65)
66          TX18= RLINE G52(18-GND-19)(PL=0.65)
67          TX19= RLINE G52(19-GND-20)(PL=0.65)
68          TX20= RLINE G52(20-GND-21)(PL=0.65)
69          TX21= RLINE G52(21-GND-22)(PL=0.65)
70          TX22= RLINE G52(22-GND-23)(PL=0.65)
71          TX23= RLINE G52(23-GND-24)(PL=0.65)
72          TX24= RLINE G52(24-GND-25)(PL=0.65)
73          TX25= RLINE G52(25-GND-26)(PL=0.65)
74          TX26= RLINE G52(26-GND-27)(PL=0.65)
75          TX27= RLINE G52(27-GND-28)(PL=0.65)
76          TX28= RLINE G52(28-GND-29)(PL=0.65)
77          TX29= RLINE G52(29-GND-30)(PL=0.65)
78          TX30= RLINE G52(30-GND-31)(PL=0.65)
79          TX31= RLINE G52(31-GND-32)(PL=0.65)
```

Figure A-7. ASTAP Model For Open Collector Driven Bus (continued)

```
 80            FEATURES
 81            GROUND = (GND)
      ----0---------1---------2---------3---------4--
 82           RLINE G52 (N1-GND-N5)


 83             ELEMENTS
 84               TO = 0.159
 85               PL = 1
 86               CMATRIX = (P11)
 87               P11 = 2.91478

      ----0---------1---------2---------3---------4---------5---------6---------7--
 88           MODEL M54FIN (IN-GND)


 89             ELEMENTS
 90             CCONN,IN-GND=4
 91             LCONN,IN-1=.015
 92             CSTUB,1-GND=6.
 93             LSTUB,1-2=.017
 94             JL,2-GND = TABLE L(VJL)
 95             CREC,2-GND = 10.

 96             FUNCTIONS
 97               TABLE L,
 98                -2.0, -121,   -0.80, -30,   -0.7,  -10,  -0.6, -02.5,
 99               -0.45,-1.10,   -0.20,-1.00,    0.2,-0.85,   0.5, -0.75,
100                1.4, -0.5,     2.0, 0.02
      ----0---------1---------2---------3---------4---------5---------6---------7--
101          MODEL MDIODE (IN-OUT)


102            ELEMENTS
103            JL,IN-OUT= TABLE L(VJL)
104            R1,IN-OUT=1000

105            FUNCTIONS
106            TABLE L,
107            .2,0,.4,1.0,.42,7,.44,9,.46,11,.5,20,.6,80,.64,110
      ----0---------1---------2---------3---------4---------5---------6---------7--
108          MODEL S54FIN (IN-GND)


109            ELEMENTS
110            CL,IN-GND = 2
111            JL,IN-GND =TABLE L(VJL)

112            FUNCTIONS
113              TABLE L,
114              -2.0, -121,   -0.80, -30,   -0.7,  -10,  -0.6, -02.5,
115             -0.45,-1.10,   -0.20,-1.00,    0.2,-0.85,   0.5, -0.75,
116              1.4, -0.5,     2.0, 0.02
      ----0---------1---------2---------3---------4---------5---------6---------7--
117          MODEL M54F2448 (OUT-GND-X-Y)
```

Figure A-7. ASTAP Model For Open Collector Driven Bus (continued)

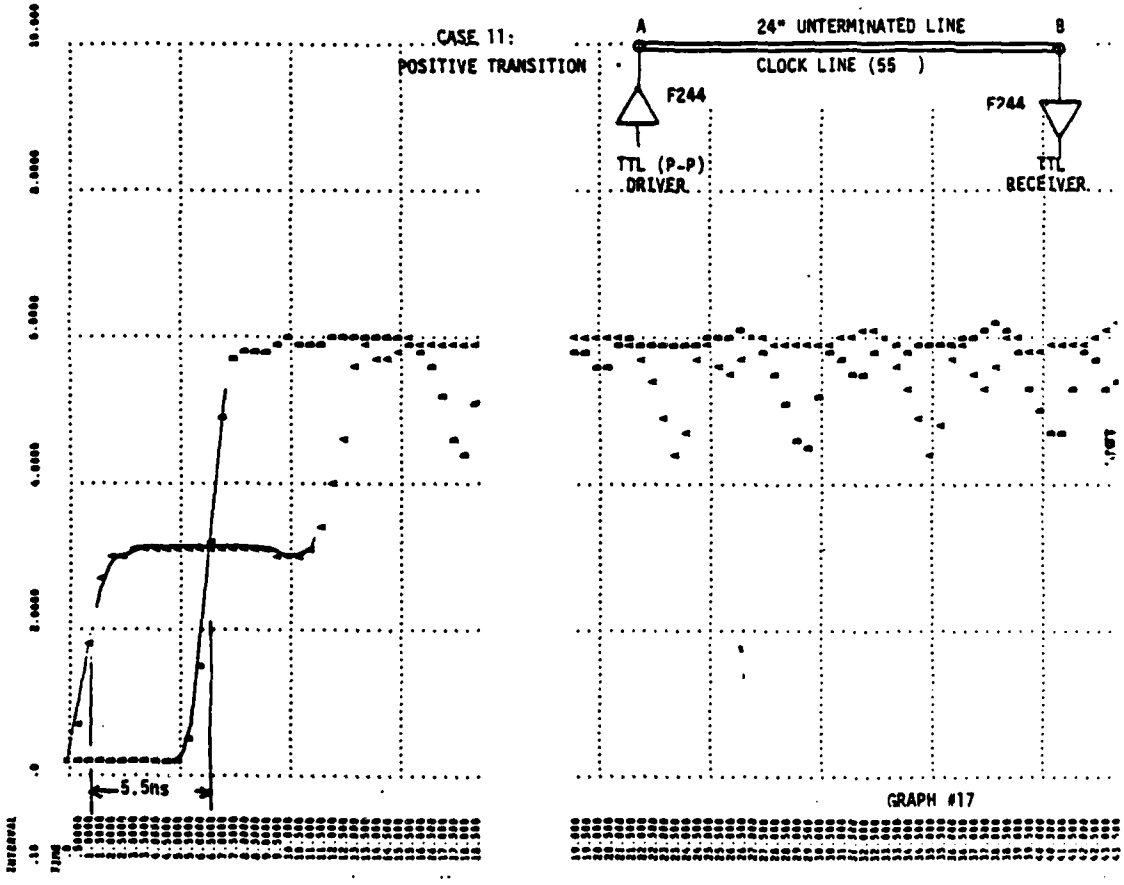
```
118             ELEMENTS
119               PR0=0.05
120               PD=1
121               A=1
122               PK = 1.1928
123               PRT = 1.2
124               PN = (PK/PRT)
125               JDX,2-GND=(((PD*(PTN-PTP))+PTP + ((1-(2*PD))*PA*(PTN-PTP)))*A)
126               CX,2-GND = 8.
127               RX,2-GND = 10E5
128               LSTUB,2-1 = .017
129               CSTUB,1-GND = 6.
130               LCONN,1-OUT = .015
131               CCONN,OUT-GND = 4
132               RIC,X-GND=10E5

133               RIW,Y-GND=10E5
134               PTN = TABLE N (VJDX)
135               PTP = TABLE P (VJDX)
136               PA =((((PTNP+PIA)*PD)+((PTPN+PIC)*(PD-1))+(PEX/PR0))/PB)
137               PTNP = TABLE N (PEY)
138               PIA = (-1*PJS)
139               PIC = (-1*PJS)
140               PEX = (((PD*(PVB-PVA))+((1-PD)*(PVC-PVD)))*(PEE))
141               PEE = (1-DEXP(-1*((PN*TIME)**2)))
142               PB = ((PD*(PTNP-PTPP)) + ((1-PD)*(PTNN-PTPN)))
143               PEY = (PVA + PEX)
144               PEZ = (PVC - PEX)
145               PTPP = TABLE P (PEY)
146               PTNN = TABLE N (PEZ)
147               PTPN = TABLE P (PEZ)
148               PVB = (VRIW)
149               PVA = (VRIC)
150               PVC = (VRIC)
151               PVD = (VRIW)

152             FUNCTIONS
153               TABLE P,
154               -2.0,-133,-.5 ,-83.3 ,-.2 ,-73.3,.0, -67, 1.0, -33.3,
155               2.0,.0
156               TABLE N,
157               -2.0, -125,   -0.77, -125, -0.699,-12.3,-0.640, -4.99,
158               0.14,-2.63, .3.4 ,.4.36,..6.88,.9,106 ,   1.2,  108,
159               1.75, 114.,    2.15,  128,   2.75,  148,   3.5,   166
```

Figure A-7. ASTAP Model For Open Collector Driven Bus (continued)

```
160              FEATURES
161               GLOBAL=(PJS)
      ----0---------1---------2---------3---------4---------5---------6----
162           MODEL S54F244B (OUT-GND)


163              ELEMENTS
164                PD= 1
165                JNA.OUT-GND=(PJN*(1-PD))
166                JPA.OUT-GND=(PJP*PD)
167                PJN=TABLE N(VJNA)
168                PJP=TABLE P(VJPA)
169                PJDX=(JNA+JPA)
170                RX.OUT-GND= 10E5
171                CX.OUT-GND= 8

172              FUNCTIONS
173                TABLE P,
174                -2.0,-133,-.5 ,-83.3 ,-.2 ,-73.3,.0, -67, 1.0, -33.3,
175                2.0, 0
176                TABLE N,
177                -2.0, -125,    -0.77, -125, -0.699,-12.3,-0.640, -4.99,
178                0.14,-2.63,     0.2, -1.0,    0.9,  223,   1.2,   238,
179                1.75,  248,    2.15,  288,   2.75,  308,   3.5,   360
      ####0#########1#########2#########3#########4#########5#########6####
      PART 2 OF THE INPUT DATA :

180           EXECUTION CONTROLS

      ----0---------1---------2---------3---------4---------5---------6-----
181           ANALYZE GETC1   (TRANSIENT)


182              RUN CONTROLS
183                START  TIME = 0
184                STOP  TIME = 50
185                PRINT INTERVAL = 1
186                STARTING STEP SIZE = .0001
187                MAXIMUM STEP SIZE = 1
188                MINIMUM STEP SIZE = .00001
189                GRAPHICS

190              INITIAL CONDITIONS
191                VCL.L1S=2.0
192                VCX.D1=2.0
193                VCX.D1S=2.0
194                VCX.D1I=.42

195              OUTPUTS
196              PRINT,PLOT (COMMON SCALES)N1,N2,N16,N32
197                 PRINT,PLOT JDX.D1, JL.L1,JL.L31
198                 PRINT          VRIC.D1, VRIW.D1, EB, PJS,PTP.D1,PTN.D1,
199                          PA.D1,PTNP.D1,PIA.D1,PEX.D1,PEE.D1,PB.D1,PTPP.D1,
200                               PTNN.D1,VR0
      ----0---------1---------2---------3---------4---------5---------6---------7--
201           END
```

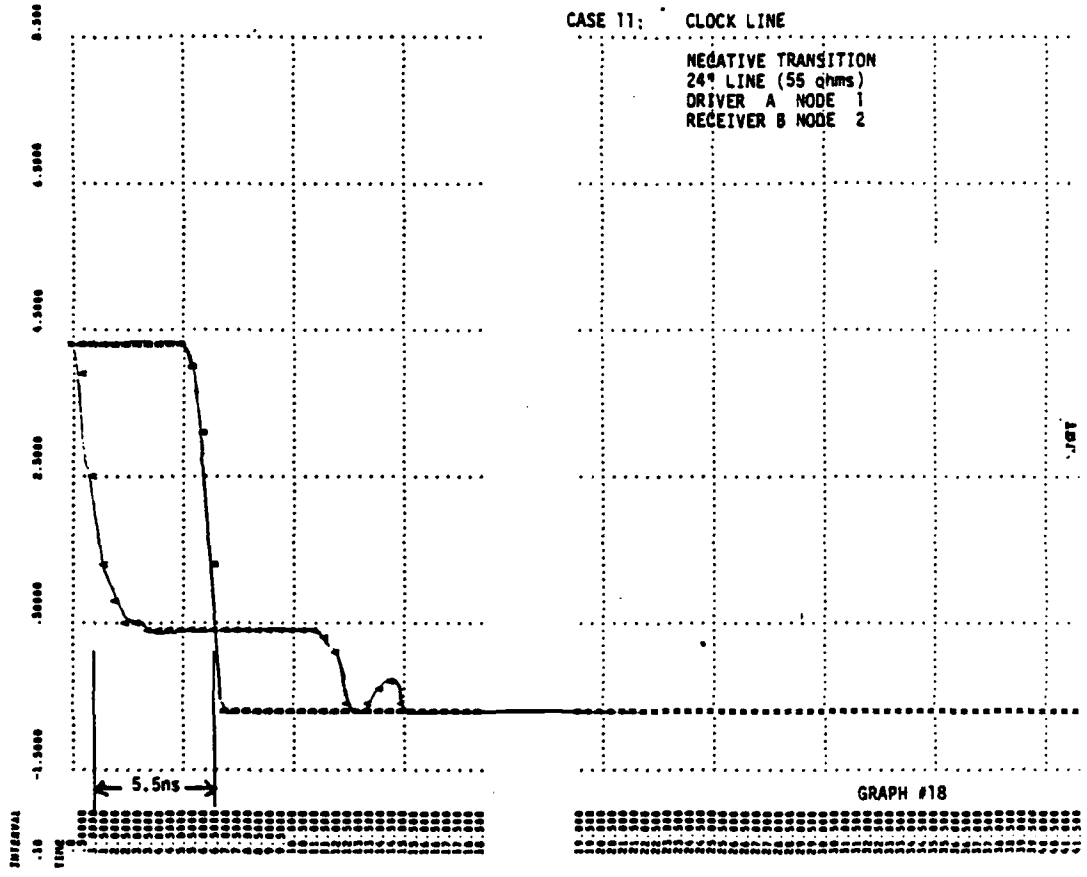
Figure A-7. ASTAP Model For Open Collector Driven Bus (continued)

To minimize clock skew, the PI-bus assumes that the bus clock is distributed directly to each module via two point nets. The electrical specification allows the use of standard TTL circuits for clock drivers and receivers on the modules. Figure A-8 and Figure A-9 show the low-to-high transition and high-to-low transition for an unterminated two point net driven by a 54F244 transceiver. In each case, the receiver at the far end of the line switches on the incident wave. The high to low transition was selected for the active edge of the bus clock because that transition is sharper and exhibits fewer reflections.



UNTERMINATED LINE
TTL 54F244 TRANSCEIVER
A - Near End Driver
B - Far End Receiver
24 inch line - 55 ohms
5 Nanoseconds Per Division

Figure A-8. 54F244 Two Point Net (Low-to-High)



UNTERMINATED LINE
TTL 54F244 TRANSCEIVER
A - Near End Driver
B - Far End Receiver
24 inch line - 55 ohms
5 Nanoseconds Per Division

Figure A-9. 54F244 Two Point Net (High-to-Low)

VHSIC Phase 2 INTEROPERABILITY STANDARDS

Appendix B

PI-Bus SPECIFICATION

March 15, 1988

Version 2.2

IBM

Honeywell

TRW

VHSIC Phase 2 INTEROPERABILITY STANDARDS

PI-Bus SPECIFICATION

VERSION 2.2

Version 2.2 is a minor update to Version 2.1 of the PI-bus Specification. The change pages are listed on the following page and the revised text is marked with a | in the left hand margin.

Change Pages Version 2.1 to 2.2

Revised Page	Comments
4-9	Clarify location of terminators
4-12	Reduce maximum MID and MIP voltage
5-8	Allow slave device to cause Bus Interface to become not selected
5-15	BIF Header (not Data) Acknowledge
5-16	Clarify acknowledge word type. Add slave device error condition for reporting software errors
5-18	Clarify error report cycle coverage
5-19	Correct spelling
5-74	Clarify Wait note a)
5-83	Correct figure reference
5-84	Correct bit field numbering
5-87	Clarify errors reported in 'Uncorrectable Line Error" bit.
5-92	Redefine slave response to reserved Message Type codes in Header Word A and allow slave device to cause Bus Interface to go not selected
5-93	Redefine slave response to reserved Message Type codes in Header Word A and add slave device error condition.
5-94	Clarify that the Suspend (S) bit is included in the error check.

CONTENTS

CONTENTS (continued)

LIST OF ILLUSTRATIONS

CONTENTS (continued)

LIST OF TABLES

ABSTRACT

This specification defines a linear, multi-drop, synchronous bus (PI-bus) which supports digital message communications between up to 32 modules residing on a single backplane. Messages are transferred datum serial and bit parallel using a datum size of 16 bits (single word) or 32 bits (double word).

The PI-bus uses a master-slave communications protocol which allows the bus master to read data from one slave or write data to any number of slaves in a single message sequence. Messages may be routed to particular modules using either logical or physical addressing. A number of independent messages may be transmitted during a bus master's tenure. The message formats provide a 32 bit virtual address range for each module.

The PI-bus protocol specifies a set of bus state transitions which control the communication sequences and allow the bus to operate in a pipelined manner at the maximum clock rate allowed by the bus signal propagation delay. Master-slave handshaking is provided with a minimal performance penalty by operating the slave modules in synchronism with the master and using bus state look-ahead.

A technique for temporarily suspending low priority block data transfers to reduce bus acquisition latency for higher priority messages is defined.

Bus mastership may be changed either by direct assignment or by priority arbitration. The protocol defines 128 logical levels of message priority and 32 levels of physical priority.

Extensive signal line and sequence error detection capability is incorporated into the bus definition. In addition, an optional single line error correction capability is specified.

PREFACE

This document was jointly prepared by IBM, Honeywell and TRW in partial fulfillment of Contract Data Requirements List (CDRL) item A011 for work being performed under VHSIC Phase 2 Submicrometer Technology Development contracts DAAK20-85-C-0367, F33615-84-C-1500 and N00039-85-C-0111, respectively.

Section 1

SCOPE

1.1 SCOPE.

This specification states the physical, electrical, functional and performance requirements defined for the PI-bus.

1.2 PURPOSE.

The purpose of this standard is to establish requirements for the PI-bus and facilitate interoperability of modules which use the PI-bus.

1.3 INTENDED APPLICATION.

The PI-bus is intended to provide a master-slave communications path for transferring digital messages between a set of up to 32 modules residing on a single backplane.

1.4 CLASSIFICATION.

Bus configurations and modules which conform to this standard may be any of the types, classes and features specified below:

Type 16	16 bit data transfers
Type 32	32 bit data transfers
Class ED	Error Detecting
Class EC	Error Correcting
Feature SO	Slave Only operation
Feature MS	Master and Slave operation

Buses and modules shall be classified according to their maximum capabilities. Bus sequences shall be classified according to the Type or Class of transfer actually used.

All modules and buses shall be capable of operating in Type 16, Class ED mode. Type 32 and Type 16 modules shall be interoperable on a Type 32 bus where the Type 32 modules may communicate using 16 or 32 bit transfers but only 16 bit transfers are used whenever a Type 16 module is an active participant. All active modules on a given bus shall operate in the same class.

Section 2

APPLICABLE DOCUMENTS

2.1 GOVERNMENT DOCUMENTS.

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- None.

2.2 NON-GOVERNMENT DOCUMENTS.

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- None.

Section 3

DEFINITIONS

The definitions listed herein shall apply to the PI-bus and PI-bus modules.

3.1 ITEM DEFINITION.

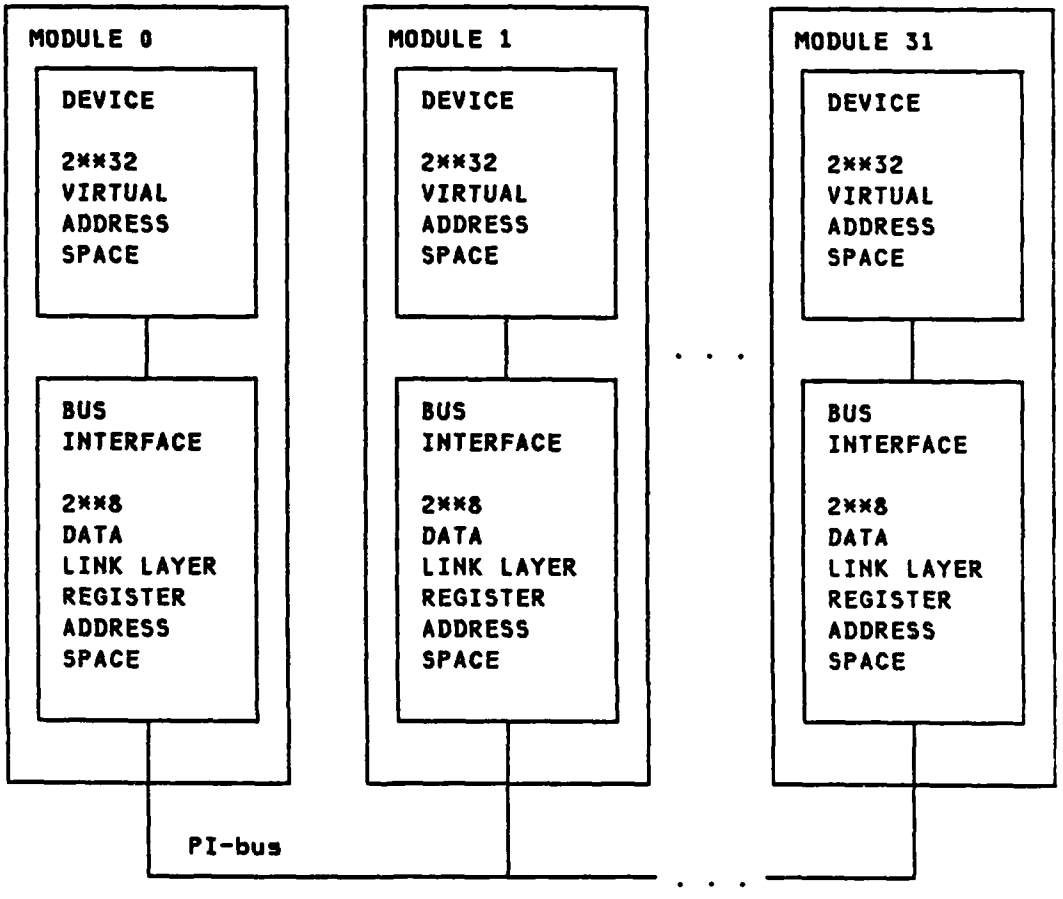
The PI-bus is a linear, multi-drop communications medium which transfers datum serial, bit parallel information among up to 32 modules residing on a single backplane. The datum size may be a single word or a double word.

PI-bus modules are those modules which implement the slave only or master and slave portions of the PI-bus protocol as specified herein.

Figure 3-1, illustrates the PI-bus and PI-bus modules. Conceptually, each module consists of a device which performs the application specific function of the module and a Bus Interface which implements the PI-bus master-slave communications protocol.

The device portion of each module is modeled as a virtual memory space with a 32 bit address range. The Bus Interface is modeled as a separate memory space with an 8 bit Data Link register address range. A separate, 8 bit virtual address called the slave ID is used by the bus master to select one or more modules to participate in a particular communications sequence as slave(s).

Figure 3-1. Conceptual Model Of Bus And Modules



MODULE VIRTUAL ADDRESS SPACE = 2**8
- PHYSICAL SLAVE ID 0 - 31
- LOGICAL SLAVE ID 32 - 255

3.2 TERM DEFINITIONS.

The definitions given below shall apply to the PI-bus and PI-bus modules.

//	The concatenation operator for groups of bits.
active Bus Interface	A Bus Interface that is connected to the bus media, and is currently capable of (and not inhibited from) participating in bus transactions.
arbitration	The process by which a single bus master is selected from competing potential bus masters.
assert (signal)	The action of changing the state of a bus signal line from released, logic 0, to asserted, logic 1, or of ensuring that the line remains in the asserted state.
asserted (signal)	The logic 1 state of a bus signal line. The least positive of the two states of a bus signal line.
backplane	A motherboard comprising wiring for the bus and connectors to the modules attached to the bus.
broadcast	A mode of operation where the bus master transmits data to all modules during a single transfer.
bus acquisition latency	The time from the highest priority module's request for bus mastership to the time at which that module becomes bus master.
bus master	The module currently in control of the bus.
bus tenure	The period between the time a bus master gains control of the bus and the time at which control is released.
contender	A potential bus master module which is actively vying for bus mastership.
device	The portion of a module, excluding the Bus Interface, which does the application depend-

ent function of the module.

double word An ordered set of 32 bits operated on as a pair of words or as a single unit. The most significant bit of a double word is labeled bit 31 and the least significant is labeled bit 0.

linear bus A bus with a single shared medium segment.

message A set of sequences starting with a header and terminating when all bus actions indicated by that header have been performed.

module An entity which is addressable via the bus and has a single connection to the bus.

multicast A mode of operation where the master transmits data to more than one slave during a single transfer.

non-transfer cycle A bus cycle that immediately follows a bus cycle in which a valid Wait is asserted or one of the VZ, HZ, DZ or HAZ cycles specified in the protocol. Information on the Data lines is not used during a non-transfer cycle.

partial message A sequence starting with a header and terminating prematurely due to a suspend, abort or other exception indication prior to a normal completion.

post (symbol) The action taken to assert and/or release individual bus lines within one particular group of bus lines such that either the associated symbol appears on the group or another symbol appears that is the result of individual line ORing of simultaneously posted symbols.

release (signal) The action of ceasing to assert a logic 1 on a bus signal line. The action of releasing a signal line produces a change in the state of the signal line only if no module is asserting that signal.

released (signal) The logic 0 state of a bus signal line produced when no module asserts the signal associated with that line. The more positive of the two states of a bus signal line relative

to the 0 Volt logic reference.

sequence A transaction comprising a number of ordered transfers performing one intended function.

slave A module which is selected by the bus master to participate in a message sequence.

symbol A unit of information on a particular group of bus lines, as represented by a particular binary encoding of bits. A valid symbol is one which conforms to the signal definitions herein including correct parity, Hamming encoding or redundant coding as applicable.

transfer A set of elemental operations on the bus which results in the communication of a bit parallel datum unit between the current bus master and the selected slave(s). The datum unit is either 16 bits (Type 16) or 32 bits (Type 32). See sequence.

word An ordered set of 16 bits operated on as a unit. The most significant bit is labeled bit 15 and the least significant bit is labeled bit 0.

Section 4

PHYSICAL LAYER

4.1 INTRODUCTION.

The physical layer of the PI-bus is specified herein. Signal lines required to implement the PI-bus are defined, including those used for signal line error detection and correction. The electrical characteristics of the module interfaces and backplane are specified and timing definitions are presented.

4.2 LINE DEFINITION.

The PI-bus signal, clock and module identification lines are defined in this section. In addition, the encoding used to achieve signal line error detection and correction is specified by definition of the valid symbols allowed for each signal line group.

4.2.1 NOMENCLATURE.

Lines shall be designated by name or by capital letter abbreviations, e.g. Data or D. Where a set of related lines are represented by the same name, the lines within the set shall be differentiated by number with the least significant bit numbered 0. The nomenclature for single lines shall be the letter abbreviation for the line name followed by the bit number enclosed in < >, e.g. D<0>. The nomenclature X<m..n> shall be the abbreviation for the set of lines Xm to Xn, inclusive, where X is the letter abbreviation for the line name and m and n are the most and least significant bit numbers, respectively. Thus, D<7..0>, represents the least significant eight Data lines. In addition, X<i,j,...,k> shall be an abbreviation for the set of lines X<i>, X<j>,..., X<k>. Thus D<5,3,1,7> stands for the set of lines D<5>, D<3>, D<1>, and D<7>.

P(X) shall designate the parity (modulo 2 sum) of the set of signal lines defined by X. Thus P(D<15..0>,DC<0>) designates the parity of the sixteen bits present on the Data lines plus the Data Check line and P(D<15..0>,DC<0>)=0 represents even parity over the specified lines.

4.2.2 BUSED SIGNAL LINES.

All PI-bus signal lines shall be implemented as wired-or lines bused between modules on a common backplane. Modules and buses shall implement those bused signal lines specified for their particular Type and Class by Table 4-1. Any implemented bus signal lines which are not required during an operation of a particular Type and Class shall be released during that operation.

Symbols posted onto signal lines shall be valid symbols as specified in this section, except that during diagnostic bus operation some invalid symbols are allowed as specified in "5.3.9.5 Diagnostics.."

As a required device function, PI-bus modules shall provide a command path independent of the PI-bus which provides a way to force all PI-bus signal lines to be released by the module. This capability may be used in bus diagnostics and fault isolation. In addition, no signal line shall be asserted from the time power is applied to the PI-bus module until the module has completed reset as defined in "5.3.8 Initialization.." Modules shall *not* assert any signal line in violation of this specification during power failure.

Table 4-1. PI-bus Signal Line Requirements By Type And Class

	Lines Required (Yes/No)			
	Type 16		Type 32	
NAME	Class ED	Class EC	Class ED	Class EC
Data (D)				
D<31..16>	No	No	Yes	Yes
D<15..0>	Yes	Yes	Yes	Yes
Data Check (DC)				
DC<7..2>	No	Yes	No	Yes
DC<1>	No	Yes	Yes	Yes
DC<0>	Yes	Yes	Yes	Yes
Cycle Type (CT)				
CT<2..0>	Yes	Yes	Yes	Yes
CT Check (CTC)				
CTC<2,1>	No	Yes	No	Yes
CTC<0>	Yes	Yes	Yes	Yes
Acknowledge Set (AS)				
AS<5,4>	No	Yes	No	Yes
AS<3..0>	Yes	Yes	Yes	Yes
Wait (W)				
W<2>	No	Yes	No	Yes
W<1,0>	Yes	Yes	Yes	Yes
Bus Request (BR)				
BR<2>	No	Yes	No	Yes
BR<1,0>	Yes	Yes	Yes	Yes
Total Lines Required	29	42	46	58

4.2.2.1 Data Line Group (D//DC)

The Data Group shall consist of the Data (D) lines and the Data Check (DC) lines. The Data Group is a set of bidirectional lines which shall transfer header, data and acknowledge information between the bus master and the slave(s). The Data Group (D//DC) lines shall also be used to resolve priority during a Vie sequence.

4.2.2.1.1 Data Lines - Type 16.

Type 16 modules and buses shall provide 16 Data lines, D<15..0>. D0 shall be the least significant and D15 the most significant line.

4.2.2.1.2 Data Lines - Type 32.

Type 32 modules and buses shall provide 32 Data lines, D<31..0>. D0 shall be the least significant and D31 the most significant line. Type 16 data transfers shall always use D<15..0>.

4.2.2.1.3 Error Protection for the Data Line Group.

The Data Line Group shall use even parity for Class ED operation and a modified Hamming Code for Class EC operation when there is a single source for the signals. When there may be multiple sources for the signals, as during vie cycles and during multiple-slave acknowledges, the Data Line Group shall use duplication for Class ED operation and triplication for Class EC operation.

4.2.2.1.3.1 Class ED Operation.

4.2.2.1.3.1.1 Single Source. During bus cycles in which a single source is specified for the Data lines, valid symbols for the Data Line Group shall have even parity.

4.2.2.1.3.1.1.1 Type 16. The module that sources the Data lines shall also source the Data Check line such that the set of symbols on D<15..0>//DC<0> satisfies P(D<15..0>//DC<0>) = 0.

4.2.2.1.3.1.1.2 Type 32. The module that sources the Data lines shall also source the Data Check lines such that the set of symbols on D<15..0>//DC<0> satisfies P(D<15..0>//DC<0>) = 0 and D<31..16>//DC<1> satisfies P(D<31..16//DC<1>) = 0.

4.2.2.1.3.1.2 Multiple Sources. During Vie and Multiple Slave Acknowledge Cycles, the Data and Data Check lines may have multiple sources. For those operations, modules shall post duplicate copies of the required symbols using Data lines D<15..8> and D<7..0> as duplicate line sets.

Usage of the Data lines for Vie and Multiple Slave Acknowledge Cycles is specified in "5.3.3.1 Vie Sequence." and "5.2.3.3.2 Multiple Slave Acknowledge.," respectively.

4.2.2.1.3.2 Class EC Operation.

4.2.2.1.3.2.1 Single Source. During bus cycles in which a single source is specified for the Data lines, valid symbols for the Data Line Group shall use modified Hamming encoding as specified below.

4.2.2.1.3.2.1.1 Type 16. The module that sources the Data lines shall also source the Data Check lines such that the set of symbols on D<15..0>//DC<5..0> satisfies:

P(DC<5>, D<15,14,13,12,11,10,9,8>) = 0

P(DC<4>, D<15,14,7,6,5,4,3,2>) = 0

P(DC<3>, D<13,12,11,7,6,5,1,0>) = 0

P(DC<2>, D<15,13,10,9,7,4,3,0>) = 0

P(DC<1>, D<12,10,8,6,4,2,1,0>) = 0

P(DC<0>, D<14,11,9,8,5,3,2,1>) = 0

4.2.2.1.3.2.1.2 Type 32. The module that sources the Data lines shall also source the Data Check lines such that the set of symbols on D<31..16>//DC<6..0> satisfies:

P(DC<6>, D<31,30,29,28,27,26,25,24,23,22,20,19,17,16>) = 0

P(DC<5>, D<31,30,29,28,27,15,14,13,12,11,10,9,8>) = 0

P(DC<4>, D<31,26,25,24,23,15,14,7,6,5,4,3,2>) = 0

P(DC<3>, D<30,26,22,21,20,19,13,12,11,7,6,5,1,0>) = 0

P(DC<2>, D<29,25,22,21,18,17,15,13,10,9,7,4,3,0>) = 0

P(DC<1>, D<28,24,20,18,17,16,12,10,8,6,4,2,1,0>) = 0

P(DC<0>, D<27,23,21,19,18,16,14,11,9,8,5,3,2,1>) = 0

4.2.2.1.3.2.2 Multiple Sources. During Vie and Multiple Slave Acknowledge Cycles the Data and Data Check lines may have multiple sources. For those operations, modules shall post triplicate copies of the required symbols using D<15..8>, D<7..0> and DC<7..0> as triplicate line sets.

Usage of the Data lines for Vie and Multiple Slave Acknowledge Cycles is specified in "5.3.3.1 Vie Sequence." and "5.2.3.3.2 Multiple Slave Acknowledge.," respectively.

4.2.2.2 Cycle Type Line Group (CT//CTC).

The Cycle Type Group shall consist of the Cycle Type (CT) and Cycle Type Check (CTC) lines. The Cycle Type Group is a set of lines onto which the bus master shall post symbols to indicate the current bus cycle type. The bus Cycle Types shall be encoded as shown in Table 4-2.

Table 4-2. PI-bus Cycle Types and Valid Symbols

Cycle Type	Abbreviation	Class ED Symbol		Class EC Symbol	
		CT<2..0>	CTC<0>	CT<2..0>	CTC<2..0>
Abort	AB	111	1	111	001
Acknowledge	A	011	0	011	110
Data	D	001	1	001	011
Header 0	H0	101	0	101	100
Header	H	010	1	010	101
Idle	I	000	0	000	000
Suspend	S	110	0	110	010
Vie	V	100	1	100	111

4.2.2.3 Acknowledge Line Set (AS).

The Acknowledge Set is a group of lines onto which the slave(s) or contenders shall post symbols to indicate synchronization or to signal uncorrectable detected errors. Valid symbols for the Acknowledge Set shall be as defined in Table 4-3.

Table 4-3. Acknowledge Line Valid Symbols

Response	Abbreviation	Class ED Code AS<3,2>	AS<1,0>	Class EC Code AS<5,4>	AS<3,2>	AS<1,0>
Acknowledge	ACK	10	10	10	10	10
Negative Ack	NAK	11	11	11	11	11
Not Selected	NS	00	00	00	00	00
Recognize	RCG	01	01	01	01	01

4.2.2.4 Wait (W) Lines.

The Wait lines shall be a set of redundant lines which the current bus master and slave(s) may assert to obtain extra non-transfer bus cycles to supply information to the bus or to accept information from the bus.

4.2.2.4.1 Class ED.

Class ED modules and buses shall provide two Wait lines, W<1,0>, that shall operate as redundant lines. Valid symbols for this case shall be W<1,0> = 00, no wait request, and W<1,0> = 11, wait requested.

4.2.2.4.2 Class EC.

Class EC modules and buses shall provide three Wait lines, W<2..0>, that shall operate as redundant lines. Valid symbols for this case shall be W<2..0> = 000, no wait request, and W<2..0> = 111, wait requested.

4.2.2.5 Bus Request (BR) Lines.

The Bus Request lines shall consist of a set of redundant lines which shall be asserted to request that the current bus master release the bus.

4.2.2.5.1 Class ED.

Class ED modules and buses shall provide two Bus Request lines, BR<1,0>, that shall operate as redundant lines. Valid symbols for this case shall be BR<1,0> = 00, no bus request, and BR<1,0> = 11, bus requested.

4.2.2.5.2 Class EC.

Class EC modules and buses shall provide three Bus Request lines, BR<2..0>, that shall operate as redundant lines. Valid symbols for this case shall be BR<2..0> = 000, no bus request, and BR<2..0> = 111, bus requested.

4.2.3 BUS CLOCK.

Bus Clock shall be a single phase clock. All bus timing shall be referenced to the high-to-low transition of the bus clock. The generation and distribution of Bus Clock is beyond the scope of this specification. However, the period of Bus Clock shall be selected to guarantee that bus timing constraints are satisfied for the bus delays and clock skews resulting from the backplane design.

4.2.4 MODULE IDENTIFICATION.

PI-bus modules shall have inputs for a set of lines that shall be hardwired on the PI-bus backplane to provide a unique module identification.

4.2.4.1 Module Identification Lines.

The set of 5 Module Identification (MID) lines shall be hardwired on the backplane and shall be used by the module as the module's physical identification code. The identification codes shall consist of an unsigned binary number in the range of 0-31, inclusive, encoded in MID<4..0>.

4.2.4.2 Module Identification Parity Line.

A Module Identification Parity (MIP) line shall be hardwired on the backplane such that MID<4..0>//MIP<0> satisfies P(MID<4..0>//MIP<0>) = 1.

4.3 ELECTRICAL REQUIREMENTS

Electrical characteristics for the PI-bus backplane and modules shall be as specified herein.

4.3.1 BACKPLANE REQUIREMENTS.

4.3.1.1 Bus Signal Line Characteristic Impedance.

PI-Bus signal lines shall have a characteristic impedance of not less than 20 ohms and not more than 50 ohms for all operating and module loading conditions.

4.3.1.2 Bus Signal Line Termination.

Signal lines shall be terminated at each electrical end of the bus to a circuit which is the Thevenin-equivalent of a terminating resistor in series with a voltage source of not less than +1.9 Volts nor more than +2.1 Volts. The value of the terminating resistance shall be between 30 and 40 ohms, inclusive.

4.3.1.3 Bus Signal Line Resistance.

The series resistance for backplane signal lines shall be limited such that the maximum voltage rise from any asserted module output to the terminating resistance at either end of the backplane is less than 100 millivolts.

4.3.1.4 Module Identification Line Resistance.

The resistance of the grounded MID and MIP lines with respect to the signal ground shall be less than 10 ohms.

4.3.1.5 Bus Clock Requirements.

4.3.1.5.1 Voltage Levels.

The low level voltage for Bus Clock shall be less than or equal to +0.55 volts. The high level voltage for Bus Clock shall be greater than or equal to +2.4 volts.

4.3.1.5.2 Rise And Fall Time.

The rise time (Tr) of the Bus Clock from 0.8 volts to 2.0 volts shall be less than 5 nanoseconds. The fall time (Tf) of the Bus Clock from 2.0 volts to 0.8 volts shall be less than 5 nanoseconds.

4.3.1.5.3 Duty Cycle.

The ratio of the Bus Clock high state duration to the bus clock period measured at 1.5 Volts shall not be less than 0.45 nor greater than 0.55.

4.3.2 MODULE REQUIREMENTS.

4.3.2.1 Bus Clock Requirements.

4.3.2.1.1 DC Requirements.

4.3.2.1.1.1 Input Capacitance. Bus Clock capacitance to logic ground shall be less than 22 picofarads.

4.3.2.1.1.2 Input Inductance. Bus Clock series inductance from the module input to the receiver of the signal shall be less than 27 nanohenries.

4.3.2.1.1.3 Bus Clock Current. The maximum current sourced by the module when the clock input voltage is +0.55 volts shall be 1.6 milliamps. The maximum current into the module when the Bus Clock voltage is +2.4 volts shall be less than 100 microamps.

4.3.2.1.1.4 High-level Input Voltage. An Bus Clock input voltage of +2.0 volts or more shall be interpreted as a high level.

4.3.2.1.1.5 Low-level Input Voltage. A Bus Clock input voltage of +0.8 volts or less shall be interpreted as a low level.

4.3.2.1.2 AC Requirements.

Modules shall operate correctly with the Bus Clock characteristics specified in "4.3.1.5 Bus Clock Requirements.."

The maximum Bus Clock frequency for the module shall be specified. The minimum Bus Clock frequency shall be zero Hertz.

All PI-bus timing shall be referenced to the high-to-low transition of Bus Clock through a voltage of 1.5 volts.

4.3.2.2 Signal Line Requirements.

4.3.2.2.1 DC Requirements.

4.3.2.2.1.1 Input Capacitance. Signal line capacitance to logic ground shall be less than 22 picofarads.

4.3.2.2.1.2 Input Inductance. Signal line series inductance from the module input to the driver or receiver of the signal shall be less than 27 nanohenries.

4.3.2.2.1.3 Leakage Current. Over the input voltage range of +0.3 volts to +2.1 volts, the absolute value of the output current for any signal line which is not being asserted by the module shall be less than 100 microamps.

4.3.2.2.1.4 Low-level Sink Current. The low-level output sink current (Iol) drive capability for signal lines shall be greater than 95 milliamps at an output voltage of 1.15 volts.

4.3.2.2.1.5 High-level Output Voltage. The high-level output voltage shall be determined by the backplane signal line termination voltage which is +1.9 to +2.1 volts. The signal line outputs shall permit wired-OR operations on the bus.

4.3.2.2.1.6 Low-level Output Voltage. The low-level output voltage (Vol) for signal lines shall be less than 1.15 volts at an input current of 95 milliamps.

4.3.2.2.1.7 High-level Input Voltage. A signal line input voltage (Vih) of +1.6 volts or more shall be interpreted as a logic 0. A signal line input which is not electrically connected to the backplane (i.e. an open line) shall be interpreted as a logic 0.

4.3.2.2.1.8 Low-level Input Voltage. A signal line input voltage (Vil) of +1.45 volts or less shall be interpreted as a logic 1.

4.3.2.2.2 AC Requirements.

4.3.2.2.2.1 Signal Line Inputs. Figure 4-1 illustrates the timing relationships specified below.

4.3.2.2.2.1.1 Set-up Time. The maximum time that each input signal is required to be uniquely above or below the input voltage threshold for a logic 0 or logic 1 prior to the high-to-low transition of the clock (set-up time, Ts) shall be specified.

4.3.2.2.2.1.2 Hold-Time. The maximum time that each input signal is required to be uniquely above or below the input voltage threshold for a logic 0 or logic 1 following the high-to-low transition of the clock (hold time, Th) shall be specified and shall not exceed the minimum propagation delay time of the module.

4.3.2.2.2.1.3 Noise Rejection. The input signal lines shall reject and the Bus Interface shall not respond to any signal pulse whose width as measured between 1.5 volts on the low-to-high transition and 1.5 volts on the high-to-low transition is less than 4 nanoseconds.

4.3.2.2.2.2 Signal Line Outputs. The following specifications shall apply when the signal line is connected to the test circuit of Figure 4-2.

4.3.2.2.2.2.1 Propagation Delay. Propagation delay shall be measured with respect to the high-to-low transition of Bus Clock as illustrated in Figure 4-3. The reference clock voltage for timing shall be +1.5 volts. The reference signal voltage for timing shall be +1.5 volts.

The minimum and the maximum propagation delay (Tpdlh) for an output sig-

nal changing from a logic 1 (low voltage) to a logic 0 (high voltage) shall be specified for each output signal line.

The minimum and the maximum propagation delay (Tpdhl) for an output signal changing from a logic 0 (high voltage) to a logic 1 (low voltage) shall be specified for each output signal line.

4.3.2.2.2.2.2 Rise And Fall Time. The rise time (Tr) of an output signal from +1.2 volts to +1.8 volts shall be less than 9 nanoseconds. The fall time (Tf) of an output signal from +1.8 volts to +1.2 volts shall be less than 9 nanoseconds.

4.3.2.3 MID And MIP Lines.

A binary 1 shall be represented by a connection to signal ground and a binary 0 shall be represented by an open circuit. Modules shall incorporate any circuits they require to sense the MID and MIP lines. The absolute current into a grounded MID or MIP line shall be less than 1 milliamp. The maximum voltage that shall exist on an open MID or MIP line shall not exceed 5.5 volts.

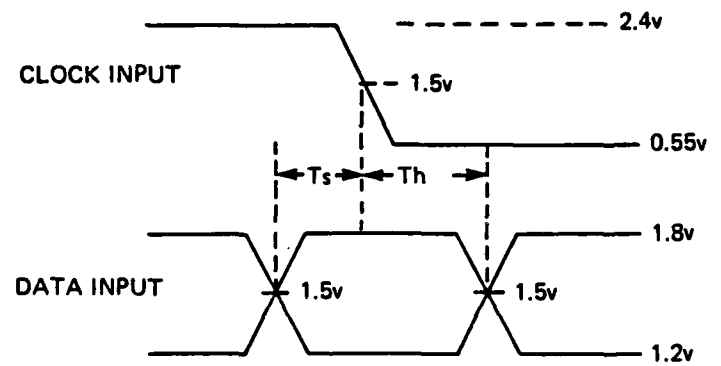


Figure 4-1. Set-up And Hold Timing

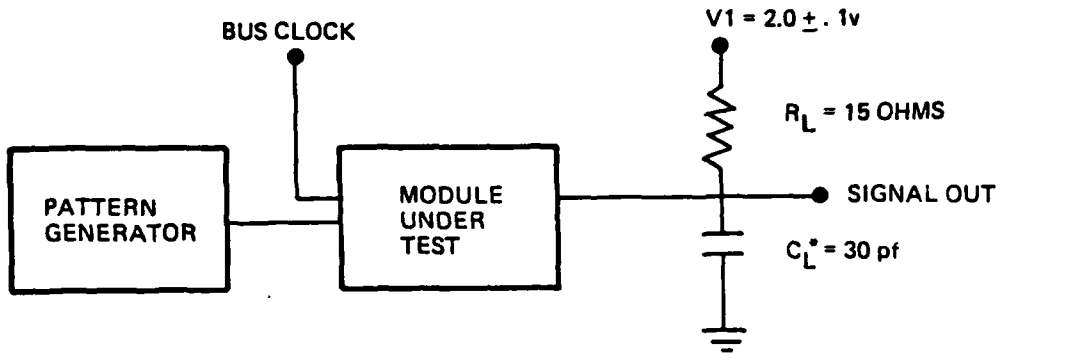


Figure 4-2. Signal Output Test Circuit

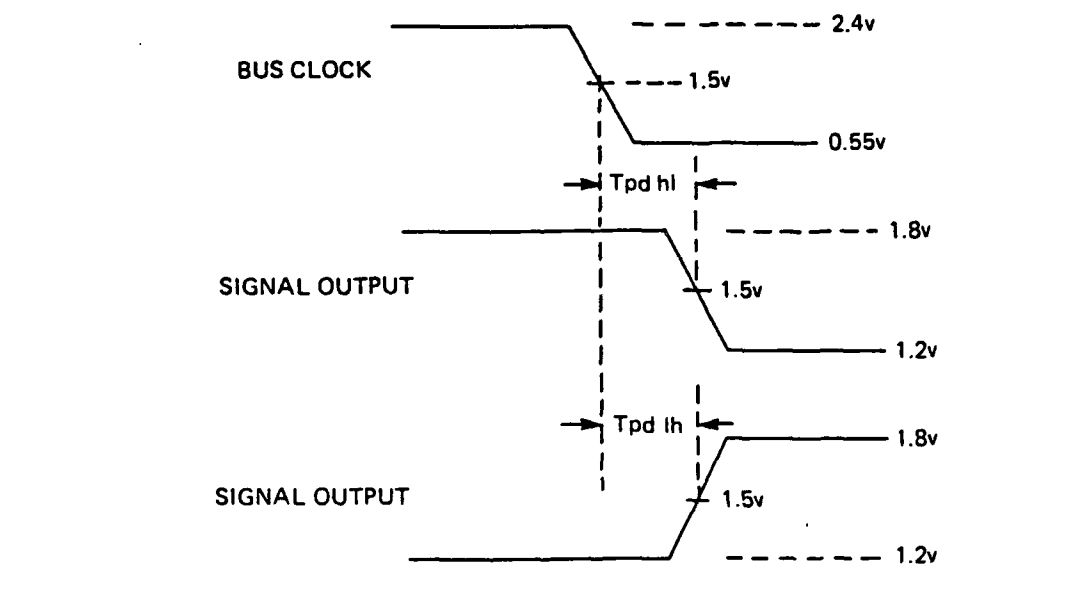


Figure 4-3. Output Signal Timing

Section 5

DATA LINK LAYER

5.1 INTRODUCTION.

The Data Link layer of the PI-bus is specified herein. The general protocol used by the PI-bus is defined through specification of the protocol state transitions and the generic message sequence. Detailed requirements for the protocol and communications sequences are specified by defining each sequence and the rules associated with the PI-bus protocol. Responses to exception conditions are defined.

5.2 GENERAL REQUIREMENTS.

5.2.1 INTRODUCTION.

The PI-bus uses a master-slave protocol under which communications sequences are defined for 1) transferring messages between modules and 2) changing bus mastership. The PI-bus communications sequences are listed in Table 5-1. The Vie sequence shall be performed only when there is no current bus master. All other sequences shall be performed under the control of the current bus master.

The PI-bus uses a set of protocol state transitions to define and control the communication sequences. Protocol state transitions shall be signaled on the Cycle Type (CT) lines and shall be controlled by the bus master. The slave(s) shall operate in synchronization with the bus master and shall signal compliance with protocol state transitions using the Acknowledge Set (AS) lines. Slave(s) shall also use the AS lines to notify the bus master of any uncorrectable errors that are detected.

The seven sequence states defined for the PI-bus protocol are summarized in Table 5-2. Within each sequence state, bus states are defined to distinguish individual bus cycles. The specific sequences of bus states required to perform PI-bus communications are defined under "5.3 DETAILED REQUIREMENTS.." In this section, general requirements for the overall operation of the PI-bus are specified by reference to the sequence states and the generic message protocol they support.

Table 5-1. PI-bus Communications Sequences

Sequence Type	Function
Mastership Sequences:	
Vie	Assigns bus mastership to the highest priority module contending for mastership through arbitration.
Tenure Pass Message	Transfers bus mastership from current bus master to another module or changes the bus master's message priority.
Message Sequences:	
Parameter Write	Transfers a 1 word parameter and a 32 bit address from the bus master device to the slave device(s).
Block Message	Transfers up to 65,536 datum units from slave device to master device or from master device to slave(s). Master sends a 32 bit address and may send 6 other Header words. May be used to continue a suspended message.
Bus Interface Message	Transfers up to 256 words from slave bus interface to the master device or from master device to slave Bus Interface(s). Master provides an 8 bit address.
Exception Sequences:	
Suspend	Suspends a Block Message data sequence and transfers Resume Control Words from the slave to the master.
Abort	Abnormally terminates current sequence.

Table 5-2. PI-bus Protocol States

Protocol State	Function
Idle	Bus not in use and no current bus master defined.
Vie	State following Idle. Used to select the next bus master from one or more contending modules. The module with the highest priority is selected as the next bus master.
Header	State in which information is transmitted by the master to specify the type of message sequence, identify modules to participate as slaves and specify additional application dependent information.
Header Acknowledge (Header Ack)	State following Header during which slave module(s) provide sequence status information to the master.
Data	State during which data are transferred between the slave module(s) and the master for Block Messages and Bus Interface Messages. Block Message Suspend sequences are performed under this protocol state.
Data Acknowledge (Data Ack)	State following Data during which slave module(s) provide message status information to the master.
Abort	State used to abnormally terminate another bus sequence.

5.2.2 PROTOCOL STATE TRANSITIONS.

The protocol states which shall be used in PI-bus operations are illustrated in Figure 5-1. All state transitions shall occur on the high-to-low transition of Bus Clock. The allowable transitions between protocol states are specified in the sections below and in Figure 5-1.

5.2.2.1 Idle.

The bus shall enter the Idle state whenever all Cycle Type lines are released. There shall be no bus master during Idle and the current bus master priority code shall be undefined. Idle shall consist of two or more consecutive bus cycles in which the Cycle Type lines are released. No PI-bus operations shall be performed during Idle except that the symbol NAK (Negative Acknowledgement) may be posted on the AS lines as specified in "5.2.3.1.2.2 Uncorrectable Errors." Vie shall be the only valid successor state to Idle. The Idle state shall be terminated and the Vie state entered only when one or more modules post the symbol V on the Cycle Type lines.

5.2.2.2 Vie.

The Vie state shall consist of eight bus cycles which shall be used to select the next bus master from one or more contenders. The Vie state shall be succeeded by the Header state except that if no bus master is selected due to erroneous operation, the bus shall return to the Idle state.

5.2.2.3 Header.

A bus master's tenure shall begin when the Header state is entered from the Vie state or from the Header Acknowledge state of the Tenure Pass message. The current bus master's tenure shall continue when the Header state is entered from the Header Acknowledge state of the Parameter Write sequence, from the Data Acknowledge state or from the Abort state. During the Header state, the bus master shall transmit header information across the bus on two or more bus cycles.

The Header shall specify the type of message sequence to be performed, identify the modules required to participate in the sequence as slaves and define the number of data transfer cycles required for the sequence. The Header state shall be succeeded by the Header Acknowledge state except that Abort may be entered to terminate the sequence.

5.2.2.4 Header Acknowledge.

The Header Acknowledge state shall be used to transmit message status from the slave module(s) to the master. The transitions out of the Header Acknowledge state shall be as specified below:

1. For a Parameter Write message sequence, the successor states to Header Acknowledge shall be Header, Idle and Abort. A transition to Header shall initiate a new message and extend the the current bus master's ten-

ure. A transition to Idle shall terminate the current bus master's tenure. Abort may be entered to terminate the Parameter Write message.

2. For Block Message and Bus Interface Message sequences, the successor states to Header Acknowledge shall be Data and Abort. A transition to Data continues the current bus master's tenure. Abort may be entered to terminate the message.

3. For a Tenure Pass Message sequence, the successor states shall be Header or Abort except that when the intended next bus master does not require or fails to acquire bus mastership, the successor state shall be Idle. The current bus master's tenure shall end at the conclusion of a Tenure Pass Message Header Acknowledge (HAZ) and the new bus master's tenure shall begin on the next cycle with entry into the Header state. Abort may be entered from a Tenure Pass Message except on the last cycle (cycle HAZ) of the message.

5.2.2.5 Data.

The Data state shall consist of a sequence of Data transfer cycles performed as part of a Block Message or Bus Interface Message. Data may be transferred from the master to the slave(s), defined as a write sequence, or from the slave to the master, defined as a read sequence. For Block Message sequences only, the Data sequence may be suspended by entry into the Suspend state. Unless a Data sequence is suspended or terminated by entering Abort, the successor state to Data shall be Data Acknowledge.

5.2.2.6 Suspend.

The Suspend state shall be used to signal the pending interruption of a Block Message Data sequence as specified in the detailed requirements (see "5.3.5.1 Suspend."). A suspended Block Message Data sequence can be resumed by another Block Message whose header contains the appropriate Resume Control Words. The successor state to Suspend shall be Data Acknowledge except that the sequence may be terminated by entering Abort.

5.2.2.7 Data Acknowledge.

The Data Acknowledge state shall be used to transfer acknowledge information from the slave(s) to the master during a Block Message or Bus Interface Message sequence. The successor states to Data Acknowledge are Header, Idle and Abort. A transition to Header shall initiate a new message and extend the the current bus master's tenure. A transition to Idle shall terminate the current bus master's tenure. Abort may be entered to terminate a message.

5.2.2.8 Abort.

The Abort state shall consist of four consecutive bus cycles in which the Abort cycle type is posted on the CT lines. The successor states to Abort shall be Header and Idle. A transition to Header shall initiate a new message and extend the the current bus master's tenure. A transition to Idle shall

terminate the current bus master's tenure.

5.2.2.9 Tenure Limitations.

5.2.2.9.1 Bus Request To Vie Interval.

When Bus Request is asserted, the bus master shall limit the number of bus cycles remaining in the current tenure to the sum of the bus cycles specified by the contents of the Vie Interval A Register plus the contents of the Vie Interval B Register plus six cycles (see "5.3.7.3.4 Vie Interval A Register - Address 3." and "5.3.7.3.5 Vie Interval B Register - Address 4."). Section "5.3.3.3 Bus Request." specifies procedures which the bus master shall use to relinquish tenure and permit a Vie sequence in response to Bus Request.

5.2.2.9.2 Absolute Tenure Limit.

PI-bus modules shall internally limit each of their individual tenures as bus master to a maximum of (2**24)+8 bus cycles. The cycle count shall begin with the first H0 cycle of the master's tenure and shall include all bus cycles (including non-transfer cycles). Each module shall provide a hardwired mechanism to automatically force all signal outputs from the module to end tenure such that this tenure limit is not exceeded. The module may resume normal operation, including vying for the bus, after allowing the bus to be in the Idle state for a minimum of two cycles.

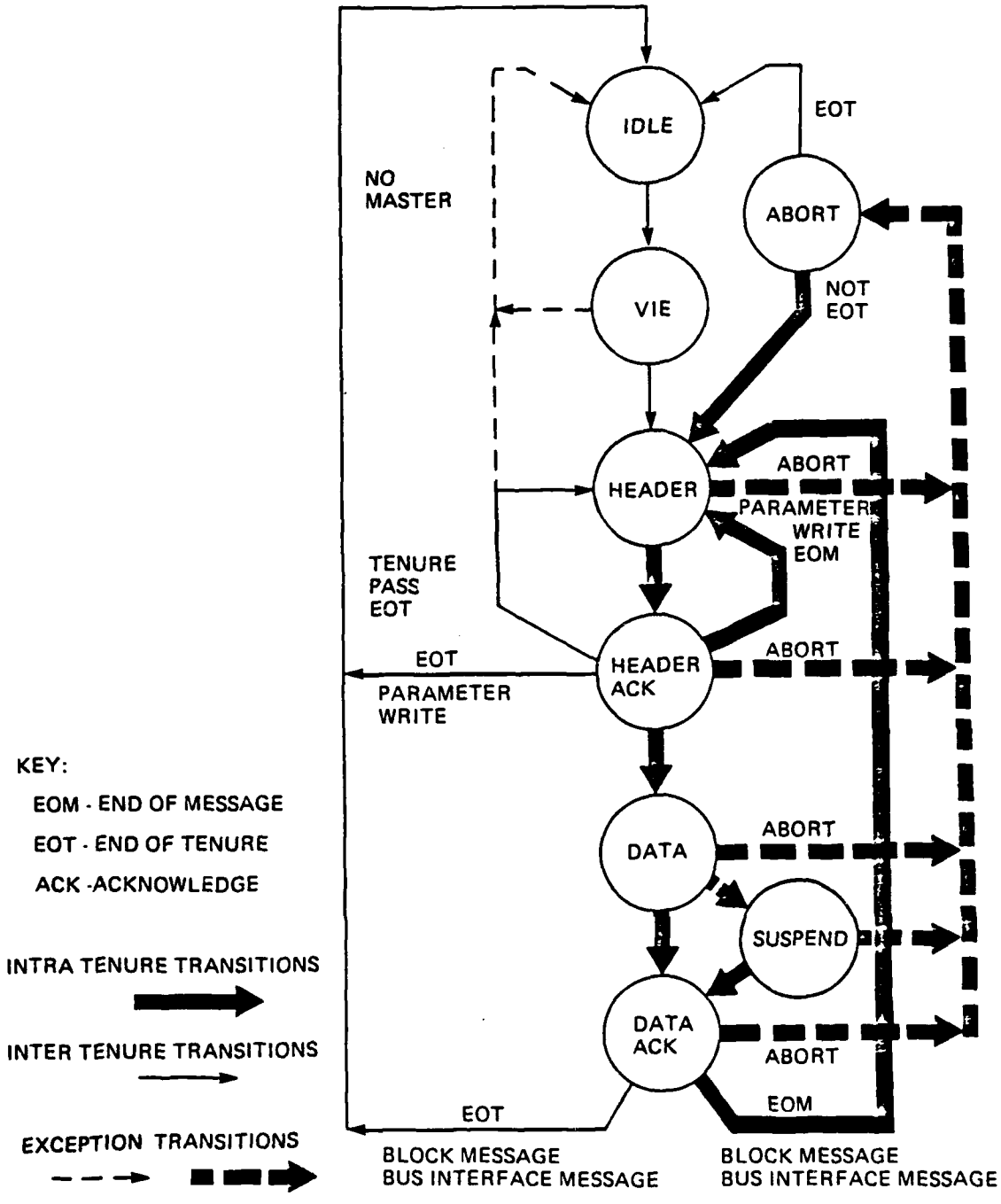


Figure 5-1. PI-bus Protocol State Diagram

5.2.3 GENERIC MESSAGE.

The generic message sequence that forms the basis for the PI-bus message sequences is described in this section. The Vie sequence is specified in "5.3.3 Bus Mastership." and the exception sequences are specified in "5.3.5 Exception Sequences.."

Table 5-3 illustrates the generic PI-bus message sequence which shall be composed of Header, Header Acknowledge, Data and Data Acknowledge sequences that correspond to the protocol states described in the preceding section.

5.2.3.1 Generic Message Sequence.

5.2.3.1.1 Normal Operation.

The Data (D) lines transfer information between the master and slave modules to accomplish the following:

1. signal the type of message sequence to be performed;

2. establish a communications path to the slave module(s);

3. transfer data between the master and the slave(s); and

4. transfer acknowledge information from the slave(s) to the master.

The bus master shall use Type 32 message sequences only when the master and all modules selected as slaves for that sequence are operating as Type 32 modules on a Type 32 bus. For a Type 32 bus, Type 32/Type 16 message sequence selection can be made on a message-by-message basis and thus may vary during the bus master's tenure.

The Cycle Type (CT) and Acknowledge Set (AS) lines shall provide handshaking between the master and slave(s) to control the sequence of bus states. The AS lines shall also be used by the slave(s) to report errors.

5.2.3.1.1.1 Header. The bus master shall initiate a message sequence by transmitting Header information on the D lines. The bus master shall post the symbol H0 on the Cycle Type (CT) lines during the first bus cycle of Header transfer and shall post H for each of the remaining bus cycles of header transfer.

The header shall specify the module(s) which are selected as slave(s) for the message sequence. Active module(s) which are addressed by the slave ID field of HWA shall become slaves on the third cycle of Header transfer. Slaves shall signal their participation in the message by posting the symbol RCG (Recognize) on the Acknowledge Set (AS) lines beginning with the third cycle of header transfer and continuing until header transfer is complete. Modules shall cease being slaves when the current message is closed by a normal completion, Abort or Suspend. Modules may also cease being slaves in response to specific slave defined conditions. All modules shall ensure that

Table 5-3. Generic PI-bus Message Sequence

	PROTOCOL BUS STATE					
SIGNAL LINES	HEADER			HEADER ACKNOWLEDGE	DATA	DATA ACKNOWLEDGE
DATA Source=	Header Master			Acknowledge Slave(s)	Data Master (Write) Slave (Read)	Acknowledge Slave(s)
CYCLE TYPE Source=Master	H0	H		A	D (S)	A
ACKNOWLEDGE Source=	NS	NS	RCG Slave	ACK Slave	RCG Slave	ACK Slave

the AS lines are released during the first two cycles of Header except that NAK shall be asserted as specified in "5.2.3.1.2.2 Uncorrectable Errors" to report errors from the preceding sequence.

5.2.3.1.1.2 Header Acknowledge. The Header Acknowledge sequence shall follow the Header sequence. A single Header Acknowledge transfer cycle shall be used for all single slave sequences. The Tenure Pass Message sequence shall include an additional (non-transfer) cycle to ensure a proper transition of mastership. The bus master shall post the Header Acknowledge Cycle (A) symbol on the CT lines during the Header Acknowledge cycle in which the slave is scheduled to post the slave Acknowledge word. The slave shall indicate synchronization with the bus master by posting ACK on the AS lines during the Header Acknowledge cycle. Five Header Acknowledge transfer cycles shall be used for a multiple slave sequence. During the first multiple slave header acknowledge cycle, all slaves post a message status symbol on the data lines and the ACK symbol on the AS lines. During each of the four remaining acknowledge cycles, eight of the thirty-two modules are assigned a bit position on the data lines upon which to post an acknowledge bit and shall indicate synchronization with the bus master by posting ACK on the AS lines.

The Header Acknowledge sequence shall complete the Tenure Pass Message and Parameter Write Message. The Block Message and Bus Interface Message sequences shall continue with a Data sequence consisting of one or more data transfer bus cycles.

5.2.3.1.1.3 Data. The bus master shall post the symbol D during each cycle of the Data sequence. The slave module(s) shall post RCG during each cycle of the Data sequence. The bus master shall transmit data during write sequences and the slave shall transmit data during the read sequences.

5.2.3.1.1.4 Data Acknowledge.. The Data sequence shall be followed by a Data Acknowledge sequence. The Data Acknowledge sequence is identical in form to the Header Acknowledge sequence. Block Messages and Bus Interface Messages shall be concluded at the end of the Data Acknowledge sequence.

5.2.3.1.2 Operation Under Exception Conditions.

5.2.3.1.2.1 Block Message Suspend. The Data sequence of a Block Message may be suspended by the bus master to permit higher priority communications. The Suspend sequence shall be performed as defined in "5.3.5.1 Suspend.."

5.2.3.1.2.2 Uncorrectable Errors. Modules which are slaves shall signal uncorrectable detected errors by posting the symbol NAK on the AS lines and providing an error log in the Acknowledge words as specified herein. Modules shall post the symbol NAK in response to an uncorrectable error which occurs during a Vie or during a Tenure Pass Message sequence.

A module that detects an uncorrectable error which applies to the operation of bus cycle N shall post the symbol NAK on bus cycle N+2. If the detected error occurred during the last two cycles of a message, the resultant NAK occurs during the first two cycles of the following message or Idle. Modules which are not slaves nor contenders in a particular message sequence shall not otherwise post NAK during that sequence.

Slave modules shall record detected error conditions for the current message in the single slave Acknowledge word or Multicast Acknowledge Register as appropriate. The Bus Interface should also notify the device of any detected errors. When an Acknowledge Word is transmitted on the bus or stored into the Multicast Acknowledge Register, the error field shall not include errors which apply to the immediately preceding bus cycle.

NAK shall have no specified effect on the resulting operation of the PI-bus other than that when NAK occurs or the asserted state of an Acknowledge word error bit is detected, the master Bus Interface should report that fact to the master device. The protocol provides the Abort sequence as a means for the message to be terminated if required by the device.

5.2.3.2 Generic Header Definition.

The PI-bus protocol defines message headers consisting of two to ten words. The first header word, Header Word A (HWA), shall be used in all message sequences to define 1) the type of message and 2) the slave modules for that message. The number of scheduled cycles in a message shall be defined by the message type and a datum transfer cycle count which shall be given implicitly in HWA or explicitly in the second header word, Header Word B (HWB). For the Bus Interface Message, HWB shall also contain an eight bit virtual address

for the Bus Interface registers. The 32 bit virtual address used in Block and Parameter Write messages shall be contained in the third and fourth header words, Header Word C0 (HWC0) and Header Word C1 (HWC1). Block Message-extended header sequences provide six additional header words, HWD0 through HWD5, for application dependent uses.

The generic format for Header Word A (HWA) is defined in this section. Formats for the remaining Header words are specific to each message sequence and are defined in "5.3 DETAILED REQUIREMENTS.."

5.2.3.2.1 Header Word A.

The format illustrated in Figure 5-2 shall be used for Header Word A. The fields of HWA shall be as specified below.

5.2.3.2.1.1 Slave Identification (ID) Field. The Slave ID field of HWA shall specify the modules required to participate in the message sequence as slaves. The Slave ID field shall provide an eight bit virtual slave address. The virtual slave address (Slave ID) range shall be partitioned into 32 single slave physical addresses, a slave broadcast address and 223 optional logical slave addresses. The broadcast Slave ID shall select all active modules as slaves, including the bus master module.

The logical Slave ID's shall be used only to define aliases for individual slave physical addresses or sets of slave physical addresses. The use of a logical Slave ID rather than a physical Slave ID in addressing a module shall not elicit any slave module response other than that which would have been produced by using the module's physical Slave ID. A Slave ID value of 0 to 31 shall specify the single module whose Module Identification matches the Slave ID field. A Slave ID value of 32 shall select all active modules as slaves. The Slave ID values 33 to 255 shall select any active modules with the given Slave ID enabled. The number of modules which respond to each of the logical Slave ID values 33 to 255 is system dependent. This means that Slave ID values 33 - 255 can be used for single slave messages or multiple slave messages depending on the application.

During message sequences where the current bus master module is selected as a slave, the module shall act as both a master and a slave to the extent that the complete standard bus message sequence can be observed on the bus lines.

5.2.3.2.1.2 Format Field (F). The Format field shall specify whether the message sequence will be performed using 16 bit (Type 16) transfers or 32 bit (Type 32) transfers. The master device must insure that 32 bit transfers are used only when no Type 16 module is selected as a slave.

5.2.3.2.1.3 Message Type Field (MSG TYPE). The Message Type field shall specify the type of message sequence to be performed according to the values in Table 5-4. The master device must ensure that only the multiple slave Message Types are used when more than one module is selected by the Slave ID field.

5.2.3.2.1.4 Access Type Field (AT). The Access Type (AT) field shall be passed from the master device to the slave module for use as defined herein and listed in Table 5-5.

Application specific AT codes may be used to classify the type of device access to be performed during a particular message. Typical uses are 1) to specify direct or indirect addressing, 2) to specify address increments, 3) to specify device dependent interpretations for extended header words and 4) to access specific processes in the device.

For Block Messages, bit 13 shall be used to signal the device that the current message is a new Block Message (bit 13 = 0) or a resumption of a previously suspended Block Message (bit 13 = 1).

For Bus Interface Messages, the code 000 shall be used to access the Data Link address space. The Bus Interface physical and Data Link layers shall not utilize any AT field information except that contained in a Bus Interface Message. Codes 001 through 011 shall be reserved for future use by higher level protocols. Higher level protocols, such as those which provide communications between modules on different backplanes, are beyond the scope of this specification.

Reserved AT codes shall be defined only by future versions of this specification.

Figure 5-2. Header Word A Format - Data Lines

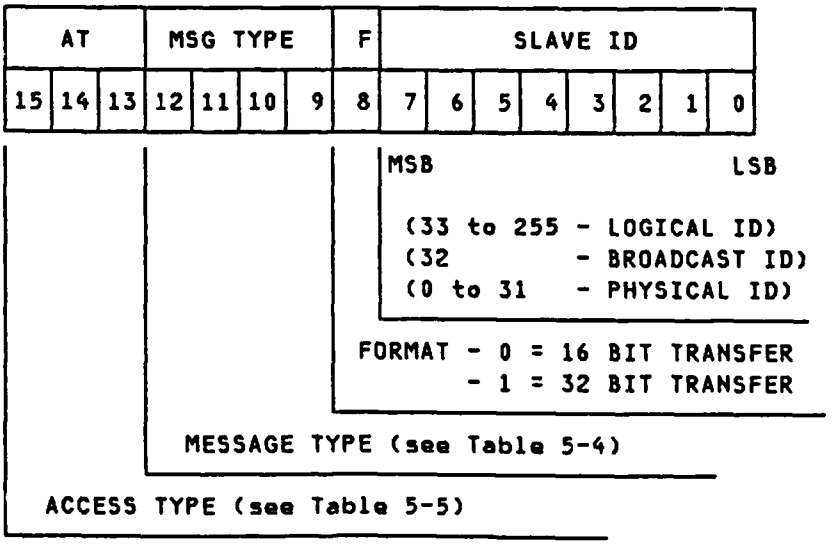


Table 5-4. Message Type Codes

MESSAGE TYPE	SINGLE OR MULTIPLE SLAVES	READ OR WRITE	MESSAGE TYPE CODE HWA <12..9>
PARAMETER WRITE	SINGLE	WRITE	0 0 0 1
	MULTIPLE	WRITE	0 0 1 1
BLOCK MESSAGE			
-SHORT HEADER	SINGLE	WRITE	0 1 0 1
	SINGLE	READ	0 1 0 0
	MULTIPLE	WRITE	0 1 1 1
-EXTENDED HEADER	SINGLE	WRITE	1 1 0 1
	SINGLE	READ	1 1 0 0
	MULTIPLE	WRITE	1 1 1 1
TENURE PASS	SINGLE	WRITE	1 0 1 0
BUS INTERFACE	SINGLE	WRITE	1 0 0 1
	SINGLE	READ	1 0 0 0
	MULTIPLE	WRITE	1 0 1 1

NOTE: Codes not listed above are reserved.

Table 5-5. Access Type Codes

SEQUENCE TYPE	ACCESS TYPE CODE HWA <15..13>	
PARAMETER WRITE	000 THRU 110	- APPLICATION SPECIFIC. (PASSED TO DEVICE)
	111	- RESERVED.
BLOCK MESSAGE		
BITS 15-14	00 THRU 11	- APPLICATION SPECIFIC (PASSED TO DEVICE)
BIT 13	0	- NEW MESSAGE
	1	- RESUME PREVIOUS MESSAGE
TENURE PASS	000	- TENURE PASS.
	001 THRU 111	- RESERVED.
BUS INTERFACE	000	- BUS INTERFACE LINK REGISTER SPACE.
	001 THRU 011	- RESERVED FOR HIGHER LEVEL PROTOCOL ACCESS.
	100 THRU 110	- IMPLEMENTATION DEFINED REGISTER SPACE.
	111	- RESERVED.

5.2.3.3 Header And Data Sequence Acknowledgement.

Header and Data Acknowledge sequences have the same form and use the same word formats. There are two basic formats for the acknowledgement, single slave and multiple slave. The single slave Acknowledge sequence shall be used when the Sequence Type field in HWA specifies a single slave sequence and the multiple slave Acknowledge sequence shall be used whenever the Sequence Type field in HWA specifies a multiple slave sequence. The single slave Acknowledge sequence transfers one word of message status information from the slave to the master. The multiple slave Acknowledge sequence transfers 1) eight bits of aggregate message status information from the slaves to the master and 2) an individual bit of message status information from each of the 32 possible slave devices to the master.

5.2.3.3.1 Single Slave Acknowledge.

The slave module shall perform error checking and logging during the message sequence. The Single Slave Acknowledge Word (AWS) defined herein provides the master with a record of the logged errors and the Module Identification of the slave.

The Single Slave Acknowledge Word shall be transmitted from the slave to the master during the Header and Data Acknowledge cycles of each single slave sequence. The master Bus Interface should pass the Single Slave Acknowledge Word to the master device.

The Single Slave Acknowledge Word format shall be as shown in Figure 5-3 and specified below.

5.2.3.3.1.1 Slave Module Identification Field (MID). The Slave Module Identification field shall contain the MID for the slave.

5.2.3.3.1.2 Acknowledge Word Type Field (AWT). The Acknowledge Word Type (AWT) field shall contain a two bit binary code as specified in Figure 5-3. An AWT code of 00 shall specify that the slave has closed the message sequence with this header or data acknowledge, as appropriate. In conjunction with an S field value of 1, an AWT of 00 shall specify message complete. In conjunction with an S field value of 0, an AWT of 00 shall specify that the Acknowledge Word completes the slave's response to a Suspend sequence as specified in "5.3.5.1 Suspend.." AWT codes 01, 10 and 11 shall specify that the slave is acknowledging the completion of the header sequence. The codes 01 and 10 further specify that the slave will respond to a Data sequence suspend with two or eight Resume Control Words, respectively. The S field code shall be 0 for an AWT code of 01 or 10. An AWT code of 11 shall further specify that the slave cannot perform a suspend sequence during the current message. The S field code shall be 1 for an AWT code of 11. The slave shall use an AWT code of 11 and an S code of 1 for any Bus Interface Message Header Acknowledge, since these messages are not suspendable.

5.2.3.3.1.3 Errors Field. The slave module shall specify detected errors in bits 7 through 13 of the Acknowledge word. Errors reported in the Header Acknowledge Word shall be those errors that originate in the current message from the start of the Header through the second cycle prior to the Header Acknowledge. Errors reported in the Data Acknowledge Word shall be those errors that originate in the current message from one cycle prior to the Header Acknowledge through the second cycle prior to the Data Acknowledge. In addition to logging current message error indications, the Bus Interface should report detected errors to the device at the time of detection.

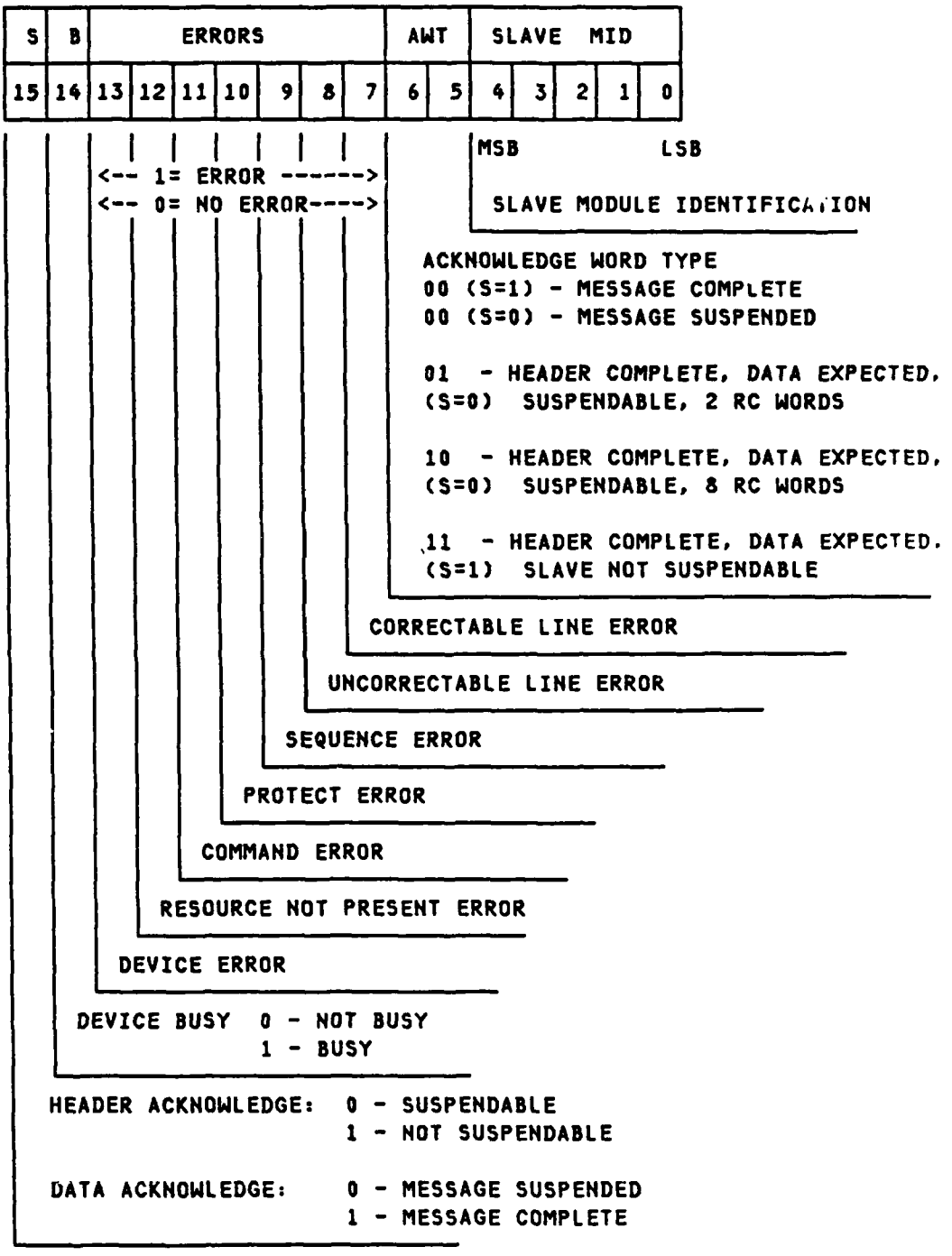
The definition of each error type in the single slave acknowledge word shall be as listed below:

Correctable Line Error	A signal line error has been detected and corrected.
Uncorrectable Line Error	A signal line error that cannot be corrected has been detected.
Sequence Error	The Cycle Type, Acknowledge Set, Wait and Bus Request line sequence of states is not in agreement with defined protocol sequences or rules.
Protect Error	A Bus Interface Message write operation has been attempted which is write protected.
Command Error	A Header Word A has been received which is not in agreement with the defined protocol or or the slave is unable to perform the commanded operation because the current bus master's priority code is unknown; or a message has been received which is not in agreement with the defined format for the slave device.
Resource Not Present Error	A resource or capability that is not implemented has been addressed in this module.
Device Error	Module device has detected an error attempting to perform a bus related operation.

5.2.3.3.1.4 Busy Field (B). The slave module shall specify in bit 14 of the Acknowledge word whether the slave device is Busy or not Busy. The device shall be recorded as Busy only when the device is unable to accept an otherwise valid message because of other operations in progress. The slave module shall post the normally scheduled acknowledge word type. The master should abort any sequence in which the slave device is specified as Busy. The master may retry the message at a later time.

5.2.3.3.1.5 Suspend Field (S). The slave shall specify in bit 15 of the Header Acknowledge Word whether the message is suspendable or not suspendable. Only Block Messages may be specified as suspendable. The slave shall specify in bit 15 of the Data Acknowledge Word whether the Data Acknowledge is in response to a Suspend sequence or the completion of the message.

Figure 5-3. Single Slave Acknowledge Word Format - Data Lines



5.2.3.3.2 Multiple Slave Acknowledge.

A Multiple Slave Acknowledge sequence shall consist of one bus transfer cycle to transmit aggregate message status and four bus transfer cycles to transmit individual acknowledge bits. The first transfer cycle shall be used by each slave to transmit a Message Status Word to the master. The individual Message Status Words shall be wire-ORed on the bus to form an aggregrate Message Status Word. The remaining four transfer cycles shall be used to transmit multiple slave Acknowledge symbols from the slave(s) to the master. The five transfer cycles are labeled HA0, HA1, HA2, HA3 and HA4 for a Multiple Slave Header Acknowledge sequence or DA0, DA1, DA2, DA3 and DA4 for a Multiple Slave Data Acknowledge sequence.

During cycles HA0 and DA0, slave modules shall transmit a Message Status word to the master using Data lines <15...0>. The format of the Message Status word shall be as defined in Figure 5-4 and Figure 5-5. Data lines <15...8> shall have the same meaning as bits <15...8> of the Single Slave Acknowledge Word and these bits shall be replicated on Data lines <7...0>, respectively, to permit error checking. If there is an uncorrectable error in the Data line group, the module shall assume that both lines in any affected pair of redundant lines are asserted. For Class EC messages, bits <15...8> shall also be replicated on Data Check lines <7...0>, respectively, to permit error correction. Modules shall not assert any Data Group line on cycle HA0 or DA0 other than the lines defined above. The Master Bus Interface should pass the aggregate Message Status to the master device.

Errors reported in the Multiple Slave Header Acknowledge word cycle HA0 shall include those errors that originate in the current message from cycle H0 through the second cycle prior to cycle HA0. During multiple slave Bus Interface Message and Block Message sequences, errors reported in the Multiple Slave Data Acknowledge word during cycle DA0 shall include those errors that originate from one cycle before HA0 through the second cycle prior to cycle DA0.

Header acknowledge cycles HA1 through HA4 shall be used to transfer "Acknowledge" multiple slave acknowledge symbols from the slave(s) to the master. Slave modules with MID values 0 through 7 shall post the "Acknowledge" symbols shown in Table 5-6 and Table 5-7 on the Data Group during cycle HA1. Slave modules with MID values 8 through 15 shall post their "Acknowledge" symbol during cycle HA2 of the sequence. Slave modules with MID values 16 through 23 shall post their "Acknowledge" symbol during cycle HA3 of the sequence and slave modules with MID values 24 through 31 shall post their "Acknowledge" symbol during cycle HA4 of the sequence. Modules shall not assert any Data Group line other than the lines included in their symbol on their assigned Acknowledge cycle.

Data acknowledge cycles DA1 through DA4 shall be used to transfer the "Acknowledge" or "No Acknowledge" multiple slave acknowledge symbols defined in Table 5-6 and Table 5-7 from the slave(s) to the master. The "Acknowledge" multiple slave Acknowledge symbol shall be posted on the Data Group during the assigned cycle of a Data Acknowledge sequence when the module is a slave and

has detected no uncorrectable errors in the current sequence. Otherwise the slave shall post "No Acknowledge" during the assigned Acknowledge cycle. Slave modules with MID values 0 through 7 shall post their multiple slave Acknowledge symbol on the Data Group during cycle DA1. Slave modules with MID values 8 through 15 shall post their multiple slave Acknowledge symbol during cycle DA2 of the sequence. Slave modules with MID values 16 through 23 shall post their multiple slave Acknowledge symbol during cycle DA3 of the sequence and slave modules with MID values 24 through 31 shall post their multiple slave Acknowledge symbol during cycle DA4 of the sequence. Modules shall not assert any Data Group line other than the lines included in their symbol on their assigned Acknowledge cycle.

The multiple slave Acknowledge symbols posted by the slave(s) during a particular Acknowledge cycle shall be logically OR'ed on the bus to produce one of the four Multiple Slave Acknowledge Words (AWM1, AWM2, AWM3 or AWM4) which shall be used during each multiple slave acknowledge sequence. The master Bus Interface should pass the Multiple Slave Acknowledge Words to the master device.

In addition to posting their assigned acknowledge symbol, slave modules shall post the symbol ACK (or NAK if required in response to an error) on the AS lines during their assigned Acknowledge cycle. Slave modules shall post the symbol NS (or NAK if required in response to an error) on the AS lines during the Acknowledge cycles in which they are not assigned to post their Acknowledge symbol.

Errors shall be logged during multiple slave sequences as specified for single slave sequences. A multicast sequence acknowledge word which has the same format and information that a Single Slave Acknowledge Word would have had if the sequence had been a single slave sequence shall be stored in the Multicast Acknowledge Register (see "5.3.7.3.1 Multicast Acknowledge Register - Address 0.") on the slave's assigned Header Acknowledge cycle. During multiple slave Bus Interface Message and Block Message sequences, a word equivalent to the Data Acknowledge word for a single slave sequence shall be formed. The acknowledge information stored in Multicast Acknowledge register bits <14..7> on the Header Acknowledge cycle shall be logically OR'ed with bits <14..7> of the equivalent single slave Data Acknowledge word and the result stored in bits <14..7> of the Multicast Acknowledge register on the slave's assigned Data Acknowledge cycle. Bit <15> and bits <6..0> of the equivalent single slave Data Acknowledge word shall be stored in Multicast Acknowledge Register bit <15> and bits <6..0>, respectively, on the slave's assigned Data Acknowledge cycle. All detected errors should be reported to the device at the time of detection.

Figure 5-4. Multiple Slave Status Word Format (AWM0) - Data Lines

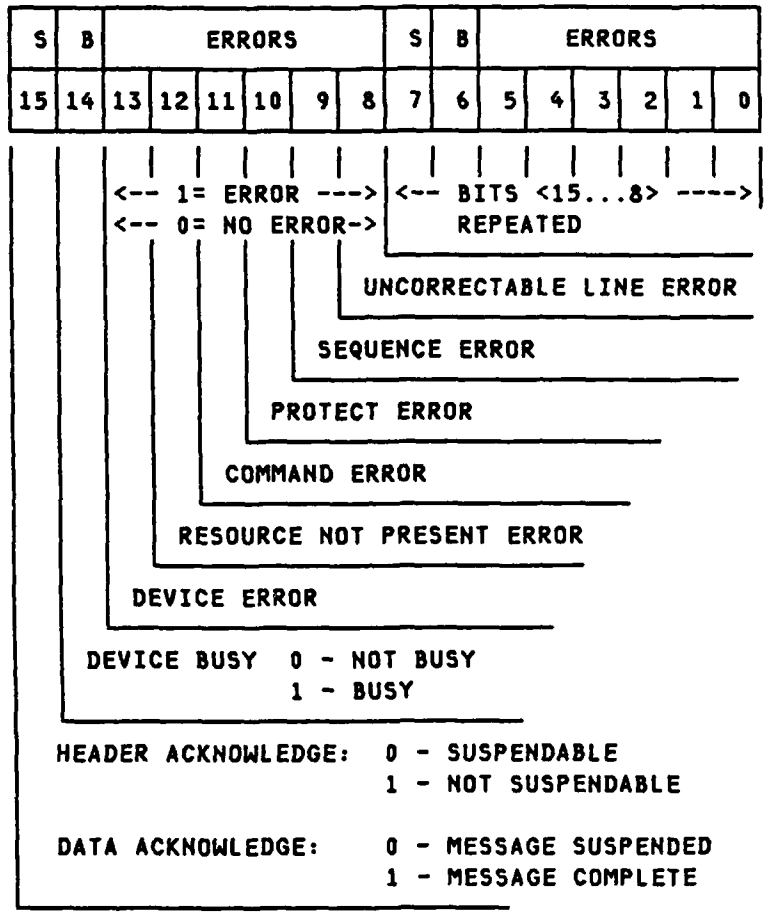


Figure 5-5. Multiple Slave Status Word Format - Data Check Lines

(Used only for Class EC)

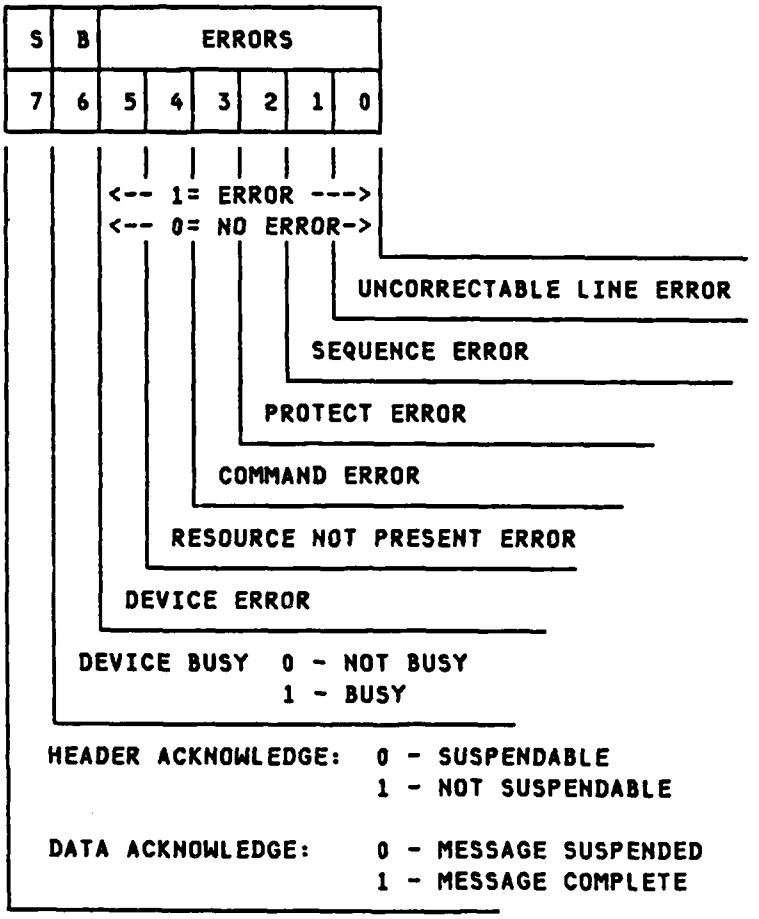


Table 5-6. Multiple Slave Acknowledge Symbol Formats (Four Words)

MODULE ID ASSIGNMENT				DATA LINES															
AWM1	AWM2	AWM3	AWM4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	8	16	24	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0	A
1	9	17	25	0	0	0	0	0	0	A	0	0	0	0	0	0	0	A	0
2	10	18	26	0	0	0	0	0	A	0	0	0	0	0	0	0	A	0	0
3	11	19	27	0	0	0	0	A	0	0	0	0	0	0	0	A	0	0	0
4	12	20	28	0	0	0	A	0	0	0	0	0	0	0	A	0	0	0	0
5	13	21	29	0	0	A	0	0	0	0	0	0	0	A	0	0	0	0	0
6	14	22	30	0	A	0	0	0	0	0	0	0	A	0	0	0	0	0	0
7	15	23	31	A	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0

A) 0=NO ACKNOWLEDGE / 1=ACKNOWLEDGE

Table 5-7. Multiple Slave Acknowledge Symbols (Four Words)

Data Check Line Formats (Used only for Class EC)

MODULE ID ASSIGNMENT				DATA CHECK LINES							
AWM1	AWM2	AWM3	AWM4	7	6	5	4	3	2	1	0
0	8	16	24	0	0	0	0	0	0	0	A
1	9	17	25	0	0	0	0	0	0	A	0
2	10	18	26	0	0	0	0	0	A	0	0
3	11	19	27	0	0	0	0	A	0	0	0
4	12	20	28	0	0	0	A	0	0	0	0
5	13	21	29	0	0	A	0	0	0	0	0
6	14	22	30	0	A	0	0	0	0	0	0
7	15	23	31	A	0	0	0	0	0	0	0

A) 0=NO ACKNOWLEDGE / 1=ACKNOWLEDGE

5.3 DETAILED REQUIREMENTS.

5.3.1 INTRODUCTION

Detailed requirements for the PI-bus Data Link protocol are specified in this section. The bus states which govern the cycle-by-cycle operation of the bus are defined and their relationships to the protocol states are given. The bus mastership protocol is specified, including requirements for the use of Bus Request. All PI-bus communications sequences are specified by sequence diagrams which show the scheduled sequence of bus states and corresponding module operations. Any sequence not specified herein shall be considered invalid.

The protocol for using Wait to insert non-transfer cycles into a message sequence is specified. The Data Link facilities which are required to be accessible over the bus are defined. Finally, bus response to error conditions is specified and bus diagnostics techniques are defined.

5.3.2 BUS STATE DEFINITIONS.

The protocol states defined in "5.2 GENERAL REQUIREMENTS." consist of sequences of bus states. The bus states shall define the information content of each bus cycle. Table 5-8 lists the bus states along with their corresponding Cycle Types and the protocol states in which they appear.

Table 5-8. Bus State Definitions

BUS STATES	CT	PROTOCOL STATE	COMMENT
I	I	Idle	Bus Idle cycle.
V0 .. V3	V	Vie	Vie priority bits resolved, 3 per step.
VZ0 .. VZ3	V	Vie	Non-Transfer cycle for vie decision time.
H0	H0	Header	First cycle of header transfer.
H1 .. H9	H	Header	Additional cycles of header transfer.
HZ	H	Header	Non-Transfer cycle for decision time.
HA0	A	Header Ack	Single slave or first multicast Header Acknowledge.
HA1 .. HA4	A	Header Ack	Additional multicast header acknowledge.
HAZ	A	Header Ack	Non-Transfer cycle for decision time.
D	D	Data	Datum transfer cycles.
DZ	D	Data	Non-Transfer cycle for decision time.
DA0	A	Data Ack	Single slave or first multicast Data Acknowledge.
DA1 .. DA4	A	Data Ack	Additional multicast acknowledge cycles.
S0 .. S2	S	Data	Cycles announcing message being suspended.
AB0 .. AB3	AB	Abort	Cycles announcing sequence being aborted.

Each of the PI-bus communication sequences defined herein has a corresponding sequence diagram which shows the required schedule of bus states and the corresponding bus operations for that sequence.

The Sequence diagrams contain the following information:

BUS STATE

Bus State Unique bus state is shown for each scheduled bus cycle.

DATA

D<31..16> Data format type or state for the most significant 16 bits of a 32 bit bus.

D<15..0> Data format type or state for the 16 bit bus or the least significant 16 bits of a 32 bit bus.

Source Bus Interface driving the data lines; master (M) or slave (S).

Read Source Shows cycles where the slave (S) sources the Data during a read sequence.

Write Source Shows cycles where the master (M) sources Data during a write sequence.

If no Source, Read Source or Write Resource is shown, data lines are released.

CYCLE TYPE

Cycle Type Defines Cycle Type symbol for each scheduled bus cycle.

Source Shows the source of the Cycle Type symbol as the master (M) or a contender (C).

ACKNOWLEDGE SET

Acknowledge Defines the state of the Acknowledge Set lines for each bus cycle. The AS lines may be driven by a single slave, by multiple slaves or by contenders in Vie. For the multiple slave case, an NS (Not Selected) beneath ACK means that an NS symbol may appear for that bus cycle if there are no sources for ACK during that cycle out of the group of potential slave (S) sources.

Source The Acknowledge Source is shown as slave(s) (S) or contender(s) (C) or, if not shown, the lines are released.

Source <7-0> For multiple slave case, the Source maybe any slave(s) (S) with an MID value of 0 thru 7.

Source <15-8> For multiple slave case, the Source maybe any slave(s) (S) with an MID value of 8 thru 15.

Source <23-16> For multiple slave case, the Source maybe any slave(s) (S) with an MID value of 16 thru 23.

Source <31-24> For multiple slave case, the Source maybe any slave(s) (S) with an MID value of 24 thru 31.

WAIT

Allowed Defines bus cycles where Wait may be asserted. A source for Wait is not shown in these sequences since the scheduled sequence of bus states assumes that Wait is not asserted.

A set of four colons (::::) in a sequence diagram indicates that a number of bus states occur in the sequence at that point.

5.3.3 BUS MASTERSHIP.

The protocol governing bus mastership is specified herein. The Vie and Tenure Pass Message sequences which assign bus mastership to a particular module are defined. The protocol which allows a module with higher priority than the current bus master to request a Vie sequence by asserting Bus Request is specified.

5.3.3.1 Vie Sequence.

The Vie sequence shall be used to determine a single bus master for the first Header sequence which occurs after the bus enters the Idle state. The bus master shall be selected on the basis of the Vie Priority code stored in each contender's Vie Priority Register (see "5.3.7.3.6 Vie Priority Register - Address 5."). The selected bus master may retain tenure or may assign tenure to another module by using the Tenure Pass Message sequence defined in the next section.

Any module that requires bus mastership may initiate Vie after two or more cycles of Idle. Modules shall be capable of participating in a Vie sequence which begins on the third or any later cycle of Idle. All active modules shall monitor each step of the Vie process and store the Vie Priority level of the winning module.

Due to pipeline delays, a module may attempt to initiate Vie up to one cycle after Vie is initiated by another module. In that case, the module which attempted to initiate the late Vie sequence shall cease to contend on next cycle and shall complete the original Vie sequence as a non-contender. Modules shall not attempt to initiate Vie more than one cycle after the V cycle type has been posted on the Cycle Type lines.

The Vie sequence shall be a four step sequence as defined in Table 5-9. Each vie step shall use two bus cycles to resolve three of the twelve bits of Vie Priority code. Modules which are contending for bus mastership shall decode the three most significant bits (VP<11..9>) of their Vie Priority code into a one-of-eight Module Vie Code as shown in Table 5-10 and Table 5-11. Modules shall initiate Vie by posting the Module Vie Code on the Data line group and the V cycle type on the Cycle Type lines. On the following bus cycle, contenders shall release the Data line group.

Table 5-9. Vie Sequence

	BUS STATE							
SIGNAL LINES	V0	VZ0	V1	VZ1	V2	VZ2	V3	VZ3
DATA * D<31..16> D<15..0> Source=	0 VC0 C	0 0	0 VC1 C	0 0	0 VC2 C	0 0	0 VC3 C	0 0
CYCLE TYPE Source=C	V	V	V	V	V	V	V	V
ACKNOWLEDGE Source=	NS	NS	RCG C	RCG C	RCG C	RCG C	RCG C	RCG C
WAIT Allowed	0 NO	0 NO	0 NO	0 NO	0 NO	0 NO	0 NO	0 NO
STEP SUB-CYCLE	1	2	1	2	1	2	1	2
VIE PRIORITY CODE BITS	11,10,9		8,7,6		5,4,3		2,1,0	
VIE STEP	0		1		2		3	

* - VCx is the vie code formed by the inclusive 'OR' of the vie codes asserted by all contending modules (C).

Modules shall read the logical-OR of the Module Vie Codes posted by the contenders from the Data line group at the end of the first cycle of Vie (bus state V0) as an Aggregate Vie Code (VC0). During the second cycle of each step, each contender (C) shall compare the posted Module Vie Code to the Aggregate Vie Code read from the bus. If the Aggregate Vie Code read from the bus has a bit asserted in a more significant bit position than the Module Vie Code posted by the module then the module has lost contention and shall not drive the bus on any remaining cycles in the Vie sequence. If the contender's posted Module Vie Code has the same bit asserted as the most significant bit asserted in the Aggregate Vie Code the module has not lost the vie step and shall proceed to the next Vie step as a contender. If there is an uncorrectable error in the Data line group, the module shall assume that both lines in any affected pair of redundant lines are asserted.

This process shall be repeated in the second, third and fourth vie steps using Vie Priority code bits VP<8..6>, VP<5..3> and VP<2..0>, respectively. At the conclusion of the fourth vie step, at most one module remains as a contender and that module shall become the new bus master. If a bus master is selected, uniqueness is guaranteed by the five MID bits within the Vie Priority Code. The bus master must post H0 on the bus cycle immediately following the VZ3 cycle. If no bus master is selected due to error conditions, the bus shall return to the Idle state.

Table 5-9 defines the detailed sequence of bus states and module actions during the Vie sequence. The Data lines are shown as two groups of sixteen lines each. Lines D<31..16>, if implemented, shall remain released throughout the Vie sequence. Contending modules shall post their Module Vie Codes on D<15..0> and, for Class EC buses only, on DC<7..0>. The Module Vie Codes posted by the contending modules shall be logically OR'ed during the first cycle of each vie step to form the Aggregate Vie Code for that step. Contenders shall post the symbol V on the Cycle Type lines during each cycle of Vie as illustrated. Contenders shall also post the symbol RCG on the AS lines during each cycle of the second, third and fourth vie steps. All active modules shall post NAK on the AS lines on cycle N+2 each time an uncorrectable Data group or Cycle Type group error which applies to the operation of bus cycle N is detected.

Modules shall not assert Wait nor Bus Request during the Vie sequence. Non-contenders shall not assert any line or post any symbol during the Vie sequence other than posting NAK on the AS lines as specified above. All active modules shall monitor the Vie sequence and record the winning bus master's priority code. Uncorrectable Data or Cycle Type group line errors detected during the Vie sequence shall cause the modules which do not win the Vie sequence to store "unknown" as the current bus master's priority code. During sub-cycle 1 of each Vie step, Data line errors for bit pairs other than the most significant pair that has a line asserted may be considered correctable on Class ED buses since the winning priority is not affected. A module which has "unknown" for the current bus master's priority code shall signal "command error" to the bus master if the module becomes a slave during the bus master's tenure (see "5.3.9 Error Detection, Recovery And Diagnostics.").

Table 5-10. Module Vie Code Format - Data Lines

	DATA LINES															
BIT PATTERN	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0 0 0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
0 0 1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
0 1 0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
0 1 1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
1 0 0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
1 0 1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
1 1 0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1 1 1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
11 10 9	Vie Priority register bits for first vie step.															
8 7 6	Vie Priority register bits for second vie step.															
5 4 3	Vie Priority register bits for third vie step.															
2 1 0	Vie Priority register bits for fourth vie step.															

Table 5-11. Module Vie Code Format - Data Check Lines

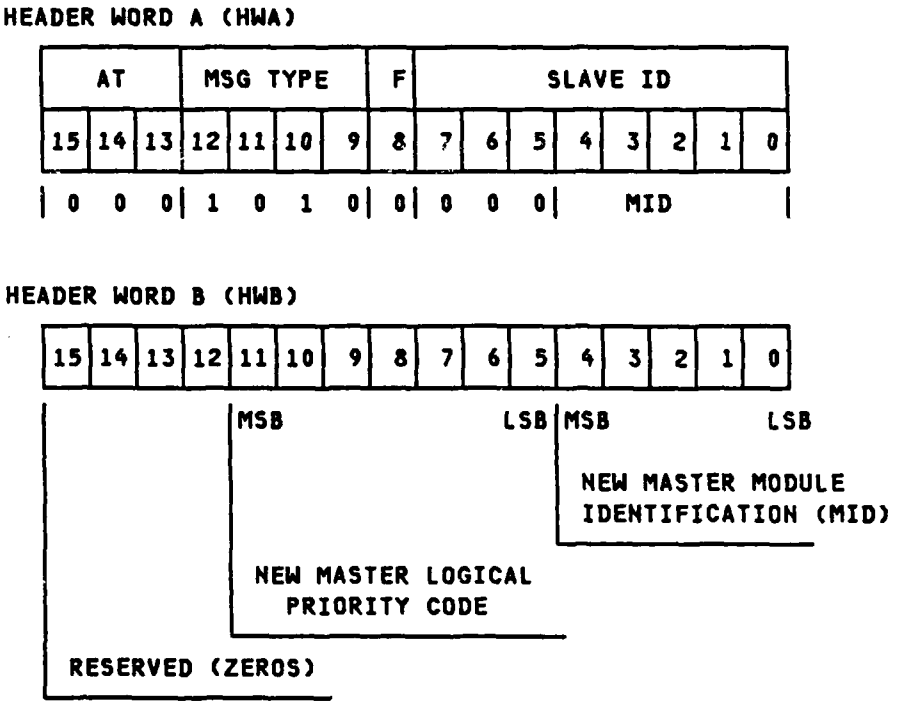
	DATA CHECK LINES							
BIT PATTERN	7	6	5	4	3	2	1	0
0 0 0	0	0	0	0	0	0	0	1
0 0 1	0	0	0	0	0	0	1	0
0 1 0	0	0	0	0	0	1	0	0
0 1 1	0	0	0	0	1	0	0	0
1 0 0	0	0	0	1	0	0	0	0
1 0 1	0	0	1	0	0	0	0	0
1 1 0	0	1	0	0	0	0	0	0
1 1 1	1	0	0	0	0	0	0	0
11 10 9	Vie Priority register bits for first vie step.							
8 7 6	Vie Priority register bits for second vie step.							
5 4 3	Vie Priority register bits for third vie step.							
2 1 0	Vie Priority register bits for fourth vie step.							

5.3.3.2 Tenure Pass Message

The current bus master may use the Tenure Pass Message to assign bus mastership directly to another module. The current bus master may also use the Tenure Pass Message to begin a new tenure under the current or a revised priority code. However, the Tenure Pass Message shall not be performed if Bus Request is in the asserted state two cycles prior to the H0 cycle on which the Tenure Pass Message sequence would have started.

The formats which shall be used for the Tenure Pass Message header words are shown in Figure 5-6. For the Tenure Pass Message, the slave ID field of Header Word A (HWA) shall be set to the Module Identification (MID) of the module which shall become the new Bus Master. The five least significant bits of HWB must be the same as the five least significant bits of HWA. Bits five through seven of HWA must be zero. The Format (F) bit in HWA shall be 0 and the Type 16 sequence shall be used for both Type 16 and Type 32 buses. Header Word B (HWB) shall contain the new bus master priority code.

Figure 5-6. Tenure Pass Message Header Word Formats



The scheduled sequence of bus states for the Tenure Pass Message shall be as shown in Table 5-12. Each bus state shall use one bus cycle except that

the master or slave may insert non-transfer cycles into the sequence by asserting Wait during the HZ and/or HA0 states.

To initiate a Tenure Pass Message, the bus master (M) shall post HWA on D<15..0> and post the symbol H0 on the Cycle Type (CT) lines during the H0 state of the Tenure Pass Message. The bus master shall post HWB on D<15..0> and the symbol H on the CT lines for the H1 state. During the HZ state, the Data group lines shall be released and the bus master shall post H on the CT lines. Also, the slave (S) shall post the symbol RCG on the Acknowledge Set group lines. On the HA0 state, the slave shall post the single slave Acknowledge word (AWS) on D<15..0> and shall post the symbol ACK on the AS group lines. Also, the master shall post the symbol A on the CT group lines. The bus master shall post A on the CT lines and the slave shall post ACK on the AS group lines during the HAZ state. The Data group lines shall be released during HAZ.

The original bus master shall not post Abort nor post Wait on HAZ and shall release all signal lines on the bus cycle following HAZ. The original bus master's tenure shall end with the HAZ state and the original slave's tenure as the new bus master shall begin on the next bus cycle.

All active modules shall monitor the Tenure Pass Message and shall acquire the new bus master's priority code from HWB. Any module that detects an error in the Tenure Pass Message shall store "unknown" as the current bus master's priority code. A module which has "unknown" for the current bus master's priority code shall signal "command error" to the bus master if the module becomes a slave during the bus master's tenure (see "5.3.9 Error Detection, Recovery And Diagnostics.").

The Tenure Pass Message shall not affect the Vie Priority Register of any module.

Table 5-12. Tenure Pass Message Sequence

	BUS STATE				
SIGNAL LINES	H0	H1	HZ	HA0	HAZ
DATA D<31..16> D<15..0> Source =	0 HWAM	0 HWBM	0 0	0 AWSS	0 0
CYCLE TYPE Source =M	H0	H	H	A	A
ACKNOWLEDGE Source =	NS	NS	RCGS	ACKS	ACKS
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 (1)

(1) Note: Only next master can assert wait on this cycle.

5.3.3.3 Bus Request.

The Bus Request line shall be asserted by a module to signal the current bus master that the module has a higher priority requirement for bus mastership. The module asserting Bus Request shall ensure that this condition is met by only asserting Bus Request when the module's Vie Priority register contains a higher priority code than that of the current bus master. A module shall not assert Bus Request when the current bus master's priority code is unknown. The current bus master shall honor Bus Request by relinquishing tenure and releasing all bus signal lines by the end of the Vie Interval defined by the sum of the bus cycles specified by the contents of the Vie Interval A Register plus the contents of the Vie Interval B Register plus six cycles (see "5.3.7.3.4 Vie Interval A Register - Address 3." and "5.3.7.3.5 Vie Interval B Register - Address 4.").

The assertion of Bus Request shall be allowed on any bus cycle, except for the cycles starting with the third cycle of Idle and continuing through the last cycle of Vie. If a module asserts Bus Request during a Tenure Pass Message and the new bus master acquires tenure with a higher priority than the module asserting Bus Request, the module shall release Bus Request within six cycles after the start of the next bus master's tenure.

The bus master shall count all bus cycles, including non-transfer cycles, whenever Bus Request is asserted. The first cycle counted shall be the first cycle after the cycle containing the initial assertion of Bus Request. Any subsequent cycle in which Bus Request is released shall not be counted and shall cause the accumulated count to be discarded. To relinquish tenure, the bus master may:

1. release all bus lines to place the bus in the Idle state rather than post an H0 cycle or

2. if a Block Message data sequence is in progress and the slave(s) have indicated in the associated Header Acknowledge that suspension is allowed, the bus master may perform a Suspend Sequence (see "5.3.5.1 Suspend.") and release all bus lines before the total Vie Interval time elapses.

If the cycle count reaches the sum of the value specified in the Vie Interval A Register plus the value specified in the Vie Interval B Register, the bus master shall perform an Abort sequence such that the first cycle of the Abort sequence occurs on the second bus cycle immediately following the cycle that exceeds the Vie Interval limit. After completing the Abort sequence the master shall immediately relinquish tenure and release all bus lines (see "5.3.5.2 Abort.").

5.3.4 MESSAGE SEQUENCES

5.3.4.1 Parameter Write Message.

The Parameter Write Message shall be used to transfer a one word parameter from the master device to a slave or multiple slave devices. The Parameter Write sequence of bus states shall be as defined in the following tables:

Type 16, single slave ---- Table 5-13

Type 16, multiple slave -- Table 5-14

Type 32, single slave ---- Table 5-15

Type 32, multiple slave -- Table 5-16

Header word formats for the Parameter Write message shall be as shown in Figure 5-7. Header Word A shall specify the slave(s), Type 16 or 32 Format, Access and Message Types. The Access Type field is application dependent, except that the code 111 shall be reserved. Message Types shall be as defined for the single slave (SS) and multiple slave (MS) cases. Header Word B shall contain the parameter information to be passed to the device. Header Word C0 and C1 shall contain 32 bits of virtual addressing information to be passed to the device.

The Header Acknowledge Word (AWS) shall supply current message status information to the master from the slave for single slave sequences. For multiple slave sequences, the Multiple Slave Status Word (AWM0) shall supply current message status information to the master. The multiple slave Header Acknowledge Words (AWM1, AWM2, AWM3, AWM4) convey the list of slave participants to the master.

Figure 5-7. Parameter Write Header Word Formats

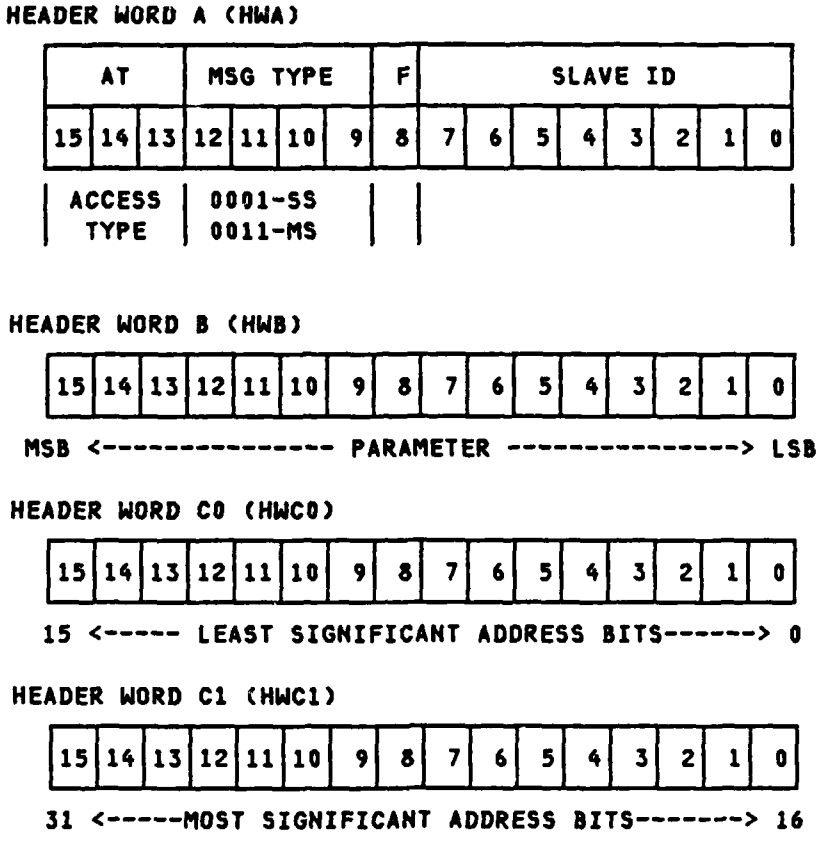


Table 5-13. Parameter Write Sequence - Type 16 Single Slave

	BUS STATE					
SIGNAL LINES	H0	H1	H2	H3	HZ	HA0
DATA D<31..16> D<15..0> Source=	0 HWA M	0 HWB M	0 HWC0 M	0 HWC1 M	0 0	0 AWS S
CYCLE TYPE Source=M	H0	H	H	H	H	A
ACKNOWLEDGE Source=	NS	NS	RCG S	RCG S	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES

Table 5-14. Parameter Write Sequence - Type 16 Multiple Slave

	BUS STATE									
SIGNAL LINES	H0	H1	H2	H3	HZ	HA0	HA1	HA2	HA3	HA4
DATA D<31..16> D<15..0> Source=	0 HWA M	0 HWB M	0 HWC0 M	0 HWC1 M	0 0	0 AWM0 S	0 AWM1 S	0 AWM2 S	0 AWM3 S	0 AWM4 S
CYCLE TYPE Source=M	H0	H	H	H	H	A	A	A	A	A
ACKNOWLEDGE	NS	NS	RCG	RCG	RCG	ACK	ACK (NS)	ACK (NS)	ACK (NS)	ACK (NS)
Source(7..0)=			S	S	S	S	S			
Source(15..8)=			S	S	S	S		S		
Source(23..16)=			S	S	S	S			S	
Source(31..24)=			S	S	S	S				S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-15. Parameter Write Sequence - Type 32 Single Slave

	BUS STATE			
SIGNAL LINES	H0	H1	HZ	HA0
DATA D<31..16> D<15..0> Source=	 HWB HWA M	 HWC1 HWC0 M	 0 0	 0 AWS S
CYCLE TYPE Source=M	H0	H	H	A
ACKNOWLEDGE Source=	NS	NS	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES

Table 5-16. Parameter Write Sequence - Type 32 Multiple Slave

	BUS STATE							
SIGNAL LINES	H0	H1	HZ	HA0	HA1	HA2	HA3	HA4
DATA D<31..16> D<15..0> Source=	 HWB HWA M	 HWC1 HWC0 M	 0 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	H0	H	H	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	NS	NS	RCG S S S S	ACK S S S S	ACK (NS) S	ACK (NS) S	ACK (NS) S	ACK (NS) S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

5.3.4.2 Block Message - Short Header Sequence.

The Block Message - Short Header (SH) Sequence shall be used to read data from a single slave device to the master device or to write data from the master device to one or more slave devices. The Block Message - Short Header sequence of bus states shall be as defined in the following tables:

Type 16, single slave ---- Table 5-17

Type 16, multiple slave -- Table 5-18

Type 32, single slave ---- Table 5-19

Type 32, multiple slave -- Table 5-20

The header word formats for the Block Message - Short Header sequence shall be as shown in Figure 5-8. Header Word A shall specify the slave(s) ID, Type 16 or 32 Format, Access and Message Types. Access Type field bits <15,14> are application dependent. Bit 13 indicates whether the message is being used to resume a previously suspended Block Message (bit 13 = 1) or send a new message (bit 13 = 0). The Message Types shall be as defined for the single slave write (SSW), single slave read (SSR) and multiple slave (MS) cases. Header Word B shall contain an unsigned binary datum count which specifies the number of datum units to be transferred between the master and slave(s), except that all zeros shall represent 65,536 datum units. The datum count shall be the number of 16 bit words for 16 bit transfers or the number of double words for 32 bit transfers. Header Word C0 and C1 shall contain 32 bits of virtual addressing information to be passed to the device. When the Block Message - Short Header is used for resuming a suspended single slave message, Header Words C0 and C1 shall be the two Resume Control Words sent from the slave to the master during the suspend sequence and they shall be returned in the order received. When the Block Message is used for resuming a suspended multicast message, the contents of C0 and C1 should be defined by application dependent convention.

The Header and Data Acknowledge Words shall supply current message status information to the master from the slave.

Data word formats are application specific. The number of words or double words transferred for each Block Message - Short Header sequence shall be equal to the value in Header Word B. For the Type 32 sequences, data words are shown with L or H designations. The least significant half of a double word or a single word transmitted on data lines D<15..0> contains the L designation. The most significant half of a double word or a single word transmitted on data lines D<31..16> contains the H designation.

Figure 5-8. Block Message - Short Header Word Formats

HEADER WORD A (HWA)

AT			MSG TYPE				F	SLAVE ID							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS TYPE			0101-SSW 0100-SSR 0111-MS												

HEADER WORD B (HWB)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MSB <---------- DATUM COUNT ----------> LSB

HEADER WORD C0 (HWC0)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

15 <----- LEAST SIGNIFICANT ADDRESS BITS------> 0

HEADER WORD C1 (HWC1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31 <-----MOST SIGNIFICANT ADDRESS BITS-------> 16

Table 5-17. Block Message - SH Sequence - Type 16 Single Slave

	BUS STATE										
SIGNAL LINES	H0	H1	H2	H3	HZ	HA0	D0	::::	Dn	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	0 HWA M	0 HWB M	0 HWC0 M	0 HWC1 M	0 0	0 AWS S	0 D0 S M	:::: :::: S M	0 Dn S M	0 0	0 AWS S
CYCLE TYPE Source=M	H0	H	H	H	H	A	D	::::	D	D	A
ACKNOWLEDGE Source=	NS	NS	RCG S	RCG S	RCG S	ACK S	RCG S	:::: S	RCG S	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES

Table 5-18. Block Message - SH Sequence - Type 16 Multiple Slave

	BUS STATE									
SIGNAL LINES	H0	H1	H2	H3	HZ	HA0	HA1	HA2	HA3	HA4
DATA D<31..16> D<15..0> Source=	0 HWA M	0 HWB M	0 HWC0 M	0 HWC1 M	0 0	0 AWM0 S	0 AWM1 S	0 AWM2 S	0 AWM3 S	0 AWM4 S
CYCLE TYPE Source=M	H0	H	H	H	H	A	A	A	A	A
ACKNOWLEDGE	NS	NS	RCG	RCG	RCG	ACK	ACK (NS)	ACK (NS)	ACK (NS)	ACK (NS)
Source(7..0)=			S	S	S	S	S			
Source(15..8)=			S	S	S	S		S		
Source(23..16)=			S	S	S	S			S	
Source(31..24)=			S	S	S	S				S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-18. Block Message - SH Sequence - Type 16 Multiple Slave (continued)

	BUS STATE								
SIGNAL LINES	D0	::::	Dn	DZ	DA0	DA1	DA2	DA3	DA4
DATA									
D<31..16>	0	::::	0	0	0	0	0	0	0
D<15..0>	D0	::::	Dn	0	AWM0	AWM1	AWM2	AWM3	AWM4
Source=	M	M	M		S	S	S	S	S
CYCLE TYPE	D	::::	D	D	A	A	A	A	A
Source=M									
ACKNOWLEDGE	RCG	::::	RCG	RCG	ACK	ACK (NS)	ACK (NS)	ACK (NS)	ACK (NS)
Source(7..0)=	S	S	S	S	S	S			
Source(15..8)=	S	S	S	S	S		S		
Source(23..16)=	S	S	S	S	S			S	
Source(31..24)=	S	S	S	S	S				S
WAIT	0	::::	0	0	0	0	0	0	0
Allowed	YES	YES	YES	YES	YES	YES	YES	YES	YES

Table 5-19. Block Message - SH Sequence - Type 32 Single Slave

	BUS STATE								
SIGNAL LINES	H0	H1	HZ	HA0	D0	::::	Dn	DZ	DA0
DATA									
D<31..16>	HWB	HWC1	0	0	D0H	::::	DnH	0	0
D<15..0>	HWA	HWC0	0	AWS	D0L	::::	DnL	0	AWS
Source=	M	M		S					S
Read Source=					S	S	S		
Write Source=					M	M	M		
CYCLE TYPE	H0	H	H	A	D	::::	D	D	A
Source=M									
ACKNOWLEDGE	NS	NS	RCG	ACK	RCG	::::	RCG	RCG	ACK
Source=			S	S	S	S	S	S	S
WAIT	0	0	0	0	0	::::	0	0	0
Allowed	NO	NO	YES	YES	YES	YES	YES	YES	YES

Table 5-20. Block Message - SH Sequence - Type 32 Multiple Slave

	BUS STATE							
SIGNAL LINES	H0	H1	HZ	HA0	HA1	HA2	HA3	HA4
DATA D<31..16> D<15..0> Source=	 HWB HWA M	 HWC1 HWC0 M	 0 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	H0	H	H	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	NS	NS	RCG S S S S	ACK S S S S	ACK (NS) S	ACK (NS) S	ACK (NS) S	ACK (NS) S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-20. Block Message - SH Sequence - Type 32 Multiple Slave (continued)

	BUS STATE								
SIGNAL LINES	D0	::::	Dn	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	D0H D0L M	:::: :::: M	DnH DnL M	0 0	0 AWM0 S	0 AWM1 S	0 AWM2 S	0 AWM3 S	0 AWM4 S
CYCLE TYPE Source=M	D	::::	D	D	A	A	A	A	A
ACKNOWLEDGE	RCG	::::	RCG	RCG	ACK	ACK (NS)	ACK (NS)	ACK (NS)	ACK (NS)
Source(7..0)=	S	S	S	S	S	S			
Source(15..8)=	S	S	S	S	S		S		
Source(23..16)=	S	S	S	S	S			S	
Source(31..24)=	S	S	S	S	S				S
WAIT Allowed	0 YES	:::: YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

5.3.4.3 Block Message - Extended Header Sequence.

The Block Message - Extended Header (EH) Sequence shall be used to read data from a single slave device to the master device or to write data from the master device to one or more slave devices. This sequence shall be used where more header information is required than that provided by the Block Message short header sequence. The Block Message - Extended Header sequence of bus states shall be as defined in the following tables:

Type 16, single slave ---- Table 5-21

Type 16, multiple slave -- Table 5-22

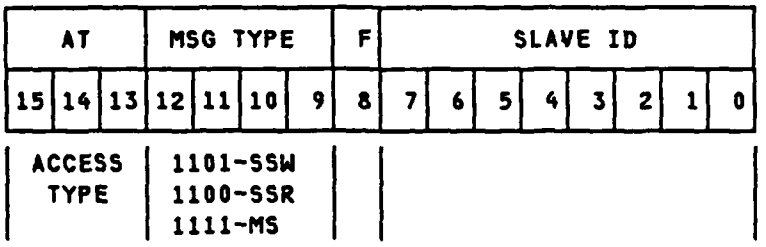
Type 32, single slave ---- Table 5-23

Type 32, multiple slave -- Table 5-24

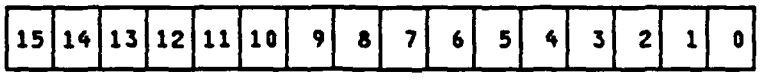
Header word formats for the Block Message - Extended Header sequence shall be as shown in Figure 5-9. Header Word A shall specify the slave(s) ID, Type 16 or 32 Format, Access and Message Types. Access Type field bits <15,14> are application dependent. Bit 13 indicates whether the message is being used to resume a previously suspended Block Message (bit 13 = 1) or send a new message (bit 13 = 0). The Message Types shall be as defined for the single slave write (SSW), single slave read (SSR) and multiple slave (MS) cases. Header Word B shall contain an unsigned binary datum count which specifies the number of datum units to be transferred between the master and slave(s), except that all zeros shall represent 65,536 datum units. The datum count shall be the number of 16 bit words for 16 bit transfers or the number of double words for 32 bit transfers. Header Word C0 and C1 shall contain 32 bits of virtual addressing information to be passed to the device. When the Block Message - Extended Header is used for resuming a suspended single slave message, Header Words C0 and C1 shall be the first two Resume Control words sent from the slave to the master during the suspend sequence and shall be returned in the order received. Header Words D0 thru D5 are also application dependent and shall be passed to the slave device. When resuming a suspended single slave message these words shall be the last six Resume Control Words sent from the slave during the suspend sequence and shall be returned in the order received. When the Block Message is used for resuming a suspended multiple slave message, the contents of HWC0, HWC1 and HWD0 through HWD5 should be defined by application dependent convention.

Figure 5-9. Block Message - Extended Header Word Formats

HEADER WORD A (HWA)

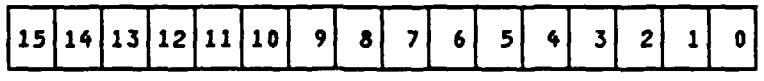


HEADER WORD B (HWB)



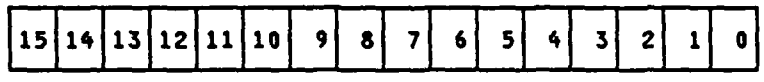
MSB <---------- DATUM COUNT ----------> LSB

HEADER WORD C0 (HWC0)



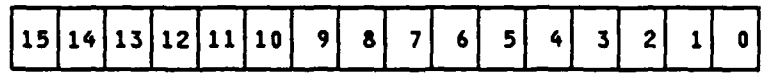
15 <----- LEAST SIGNIFICANT ADDRESS BITS------> 0

HEADER WORD C1 (HWC1)



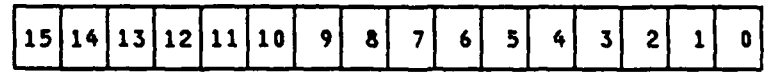
31 <-----MOST SIGNIFICANT ADDRESS BITS-------> 16

HEADER WORD D0 (HWD0)



MSB LSB

HEADER WORD D5 (HWD5)



MSB LSB

Table 5-21. Block Message - EH Sequence - Type 16 Single Slave

	BUS STATE								
SIGNAL LINES	H0	H1	H2	H3	H4	::::	H9	HZ	HA0
DATA D<31..16> D<15..0> Source=	0 HWA M	0 HWB M	0 HWC0 M	0 HWC1 M	0 HWD0 M	:::: :::: M	0 HWD5 M	0 0	0 AWS S
CYCLE TYPE Source=M	H0	H	H	H	H	::::	H	H	A
ACKNOWLEDGE Source=	NS	NS	RCG S	RCG S	RCG S	:::: S	RCG S	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-21. Block Message - EH Sequence - Type 16 Single Slave (continued)

	BUS STATE						
SIGNAL LINES	D0	D1	D2	::::	Dn	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	 0 D0 S M	 0 D1 S M	 0 D2 S M	 :::: :::: S M	 0 Dn S M	 0 0 	 0 AWS S
CYCLE TYPE Source=M	D	D	D	::::	D	D	A
ACKNOWLEDGE Source=	RCG S	RCG S	RCG S	:::: S	RCG S	RCG S	ACK S
WAIT Allowed	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES

Table 5-22. Block Message - EH Sequence - Type 16 Multiple Slave

	BUS STATE										
SIGNAL LINES	H0	H1	H2	H3	H4	::::	H9	HZ	HA0	HA1	HA2
DATA D<31..16> D<15..0> Source=	0 HWA M	0 HWB M	0 HWC0 M	0 HWC1 M	0 HWD0 M	:::: :::: M	0 HWD5 M	0 0	0 AWM0 S	0 AWM1 S	0 AWM2 S
CYCLE TYPE Source=M	H0	H	H	H	H	::::	H	H	A	A	A
ACKNOWLEDGE	NS	NS	RCG	RCG	RCG	::::	RCG	RCG	ACK	ACK (NS)	ACK (NS)
Source(7..0)=			S	S	S	S	S	S	S	S	
Source(15..8)=			S	S	S	S	S	S	S		S
Source(23..16)=			S	S	S	S	S	S	S		
Source(31..24)=			S	S	S	S	S	S	S		
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-22. Block Message - EH Sequence - Type 16 Multiple Slave (continued)

	BUS STATE										
SIGNAL LINES	HA3	HA4	D0	::::	Dn	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	0 AWM3 S	0 AWM4 S	0 D0 M	:::: :::: M	0 Dn M	0 0	0 AWM0 S	0 AWM1 S	0 AWM2 S	0 AWM3 S	0 AWM4 S
CYCLE TYPE Source=M	A	A	D	::::	D	D	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	ACK (NS) S 	ACK (NS) S	RCG S S S S	:::: S S S S	RCG S S S S	RCG S S S S	ACK S S S S	ACK (NS) S 	ACK (NS) S 	ACK (NS) S 	ACK (NS) S
WAIT Allowed	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-23. Block Message - EH Sequence - Type 32 Single Slave

	BUS STATE						
SIGNAL LINES	H0	H1	H2	H3	H4	HZ	HA0
DATA D<31..16> D<15..0> Source=	 HWB HWA M	 HWC1 HWC0 M	 HWD1 HWD0 M	 HWD3 HWD2 M	 HWD5 HWD4 M	 0 0	 0 AWS S
CYCLE TYPE Source=M	H0	H	H	H	H	H	A
ACKNOWLEDGE Source=	NS	NS	RCG S	RCG S	RCG S	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-23. Block Message - EH Sequence - Type 32 Single Slave (continued)

	BUS STATE						
SIGNAL LINES	D0	D1	D2	::::	Dn	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	 D0H D0L S M	 D1H D1L S M	 D2H D2L S M	 :::: :::: S M	 DnH DnL S M	 0 0 	 0 AWS S
CYCLE TYPE Source=M	D	D	D	::::	D	D	A
ACKNOWLEDGE Source=	RCG S	RCG S	RCG S	:::: S	RCG S	RCG S	ACK S
WAIT Allowed	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES

Table 5-24. Block Message - EH Sequence - Type 32 Multiple Slave

	BUS STATE										
SIGNAL LINES	H0	H1	H2	H3	H4	HZ	HA0	HA1	HA2	HA3	HA4
DATA D<31..16> D<15..0> Source=	 HWB HWA M	 HWC1 HWC0 M	 HWD1 HWD0 M	 HWD3 HWD2 M	 HWD5 HWD4 M	 0 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	H0	H	H	H	H	H	A	A	A	A	A
ACKNOWLEDGE	NS	NS	RCG	RCG	RCG	RCG	ACK	ACK (NS)	ACK (NS)	ACK (NS)	ACK (NS)
Source(7..0)=			S	S	S	S	S	S			
Source(15..8)=			S	S	S	S	S		S		
Source(23..16)=			S	S	S	S	S			S	
Source(31..24)=			S	S	S	S	S				S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-24. Block Message - EH Sequence - Type 32 Multiple Slave (continued)

	BUS STATE								
SIGNAL LINES	D0	::::	Dn	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	D0H D0L M	:::: :::: M	DnH DnL M	0 0	0 AWM0 S	0 AWM1 S	0 AWM2 S	0 AWM3 S	0 AWM4 S
CYCLE TYPE Source=M	D	::::	D	D	A	A	A	A	A
ACKNOWLEDGE	RCG	::::	RCG	RCG	ACK	ACK (NS)	ACK (NS)	ACK (NS)	ACK (NS)
Source(7..0)=	S	S	S	S	S	S			
Source(15..8)=	S	S	S	S	S		S		
Source(23..16)=	S	S	S	S	S			S	
Source(31..24)=	S	S	S	S	S				S
WAIT Allowed	0 YES	:::: YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

5.3.4.4 Bus Interface Message Sequence

The Bus Interface Message shall be used to read or write to the Bus Interface register address spaces. The Bus Interface Message sequence of bus states shall be as defined in the following tables:

Type 16, single slave ---- Table 5-25

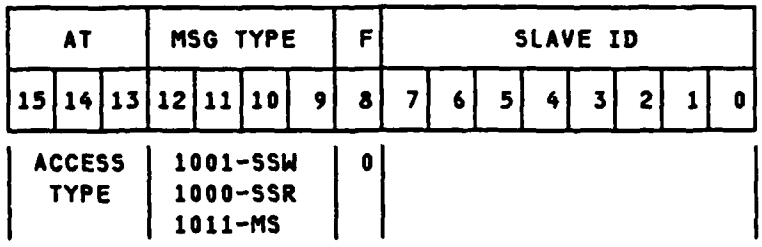
Type 16, multiple slave -- Table 5-26

Header word formats for the Bus Interface Message sequence shall be as shown in Figure 5-10. Header Word A shall specify the participant slave(s), Type 16 Format, Access and Message Types. The same format shall be used for both Type 16 and Type 32 buses. Access Type code 000 is defined for the Data Link register address space. Codes 001 through 011 shall be reserved for higher level protocols and the code 111 shall be reserved for future use. The other codes may be used for implementation defined registers. The Message Types shall be as defined for the single slave write (SSW), single slave read (SSR) and multiple slave (MS) cases. Header Word B shall contain the address for the first Bus Interface register to be accessed in the sequence and an unsigned binary word count specifying the number of data words to be transferred between the master and the slave(s), except that a word count of all zeros shall mean 256 words. Each data transfer after the first data transfer shall access a successive register address. The register address shall be incremented after each data word transfer.

Data word formats are as defined in "5.3.7 Data Link Facilities.." The number of data words transferred shall equal the value in Header Word B.

Figure 5-10. Bus Interface Message Header Word Formats

HEADER WORD A (HWA)



HEADER WORD B (HWB)

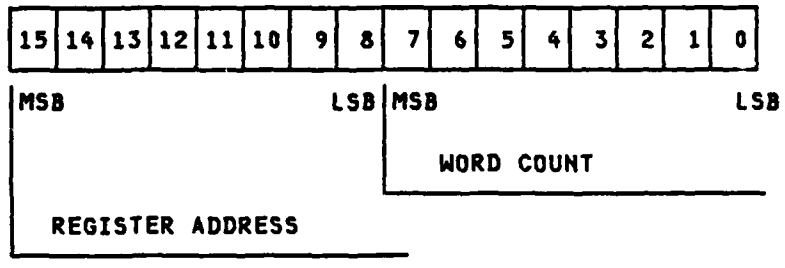


Table 5-25. Bus Interface Message Sequence - Type 16 Single Slave

	BUS STATE								
SIGNAL LINES	H0	H1	HZ	HA0	D0	: : : :	Dn	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	0 HWA M	0 HWB M	0 0	0 AWS S	0 D0 S M	: : : : : : : : S M	0 Dn S M	0 0	0 AWS S
CYCLE TYPE Source=M	H0	H	H	A	D	: : : :	D	D	A
ACKNOWLEDGE Source=	NS	NS	RCG S	ACK S	RCG S	: : : : S	RCG S	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	: : : : YES	0 YES	0 YES	0 YES

Table 5-26. Bus Interface Message Sequence - Type 16 Multiple Slave

	BUS STATE							
SIGNAL LINES	H0	H1	HZ	HA0	HA1	HA2	HA3	HA4
DATA D<31..16> D<15..0> Source=	0 HWA M	0 HWB M	0 0	0 AWM0 S	0 AWM1 S	0 AWM2 S	0 AWM3 S	0 AWM4 S
CYCLE TYPE Source=M	H0	H	H	A	A	A	A	A
ACKNOWLEDGE	NS	NS	RCG	ACK	ACK (NS)	ACK (NS)	ACK (NS)	ACK (NS)
Source(7..0)=			S	S	S			
Source(15..8)=			S	S		S		
Source(23..16)=			S	S			S	
Source(31..24)=			S	S				S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-26. Bus Interface Message Sequence - Type 16 Multiple Slave (continued)

	BUS STATE								
SIGNAL LINES	D0	::::	Dn	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	0 D0 M	:::: :::: M	0 Dn M	0 0	0 AWM0 S	0 AWM1 S	0 AWM2 S	0 AWM3 S	0 AWM4 S
CYCLE TYPE Source=M	D	::::	D	D	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	RCG S S S S	:::: S S S S	RCG S S S S	RCG S S S S	ACK S S S S	ACK (NS) S	ACK (NS) S	ACK (NS) S	ACK (NS) S
WAIT Allowed	0 YES	:::: YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

5.3.5 EXCEPTION SEQUENCES.

5.3.5.1 Suspend.

The current bus master may elect to suspend a Block Message during the Data sequence providing that there are at least three Data Cycles remaining and the corresponding Header Acknowledge Word had an S of 0 and, for single slave messages, an AWT of 01 or 10. After completing the Suspend sequence, the current bus master may initiate a new message with an H0 cycle or release the bus to initiate the Idle state.

The Suspend Sequence may be used in conjunction with the Vie Interval A and Vie Interval B Registers to optimize the trade-off between short bus acquisition latency and large message sizes by allowing a Block Message to be interrupted and resumed at a later time.

5.3.5.1.1 Single Slave Suspend.

The Suspend sequence of bus states for single slave Block Messages shall be as defined in the following tables:

Type 16, single slave, short data ---- Table 5-27

Type 32, single slave, short data ---- Table 5-28

Type 16, single slave, extended data - Table 5-29

Type 32, single slave, extended data - Table 5-30

The Short Data Sequence shall be used if the AWT field of the Single Slave Acknowledge Word from the suspended message was 01. The Extended Data Sequence shall be used if the AWT field was 10. In terms of the data transfer operation, the S cycles of these sequences shall be considered normal D cycles and all Bus Interfaces shall respond accordingly. The slave shall post ACK during the last of the three S cycles to indicate that the Suspend Sequence has been recognized.

Beginning on the fourth cycle of a Suspend sequence, the slave shall transmit to the bus master implementation/device specific Resume Control Words that the master shall store for later use in resuming the message. The Suspend - Short Data sequences transfer two Resume Control Words and the Suspend - Extended Data sequences transfer eight Resume Control Words. The Resume Control Words shall be followed by a Data Acknowledge Word.

5.3.5.1.2 Multiple Slave Suspend.

The Suspend sequence of bus states for multiple slave Block Messages shall be as defined in the following tables:

Type 16, multiple slave ---- Table 5-31

Type 32, multiple slave ---- Table 5-32

In terms of the data transfer operation, the S cycles of these sequences shall be considered normal D cycles and all Bus Interfaces shall respond accordingly. The slaves shall post ACK during the last of the three S cycles to indicate that the Suspend Sequence has been recognized. A non-transfer DZ cycle and a multiple slave Data Acknowledge sequence shall follow the third S cycle. No Resume Control Words are sent from the slaves to the master.

5.3.5.1.3 Resuming Suspended Messages

A master shall resume a suspended Block Message by transmitting the remaining data to the slave(s) in a Block Message with bit 13 of the HWA AT field set to 1. The Resume Control Words that the slave(s) may require to resume that message shall be sent to the slave(s) during the appropriate header cycles as described in "5.3.4.2 Block Message - Short Header Sequence." and "5.3.4.3 Block Message - Extended Header Sequence.." For a single slave message, the Resume Control Words shall be those words transmitted from the slave to the master during the suspend sequence. For a multiple slave message, the Resume Control Words must be created by the master according to the slaves' resume control requirements.

A Block Message - Short Header shall be used to resume a suspended single slave Block Message which had an AWT field of 01 (two Resume Control Words) and a Block Message - Extended Header shall be used to resume a suspended single slave Block Message which had an AWT field of 10 (eight Resume Control Words). The type of Block Message used to resume a multiple slave message shall be determined by the slaves' resume control word requirements.

Note that the choice of a short or extended header Block Message for the resume sequence is not determined by whether a short or extended header was used in the suspended Block Message.

5.3.5.1.4 Using Suspend With Vie Intervals A and B

The PI-bus Vie Interval A and Vie Interval B Registers may be used by some bus interfaces to determine when a Suspend sequence is required to meet the Vie Interval requirement and to limit the number of non-transfer cycles caused by assertion of Wait during a Suspend sequence. Vie Interval A may be selected to define a checkpoint where the master bus interface determines whether or not to suspend an incomplete Block Message. Vie Interval B may be selected to define the number of bus cycles allowed for completion of the Suspend sequence, including non-transfer cycles that may be required as a result of Waits. If the actual number of Waits causes the total bus cycles to exceed the number specified by the contents of the Vie Interval A Register plus the contents of the Vie Interval B Register, the master shall perform an Abort sequence such that the first cycle of the Abort sequence occurs on the second bus cycle immediately following the cycle that exceeds the Vie Interval limit. This Abort cycle shall be the first cycle in the Abort Sequence, described in "5.3.5.2 Abort.." At the end of the Abort sequence the master shall immediately release the bus to the Idle state.

Table 5-27. Suspend - Short Data Sequence - Type 16 Single Slave

	BUS STATE						
SIGNAL LINES	Di	Di+1	Di+2	RD0	RD1	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	 0 Di S M	 0 Di+1 S M	 0 Di+2 S M	 0 RCD0 S	 0 RCD1 S	 0 	 0 AWS S
CYCLE TYPE Source=M	S	S	S	D	D	D	A
ACKNOWLEDGE Source=	RCG S	RCG S	ACK S	RCG S	RCG S	RCG S	ACK S
WAIT Allowed Source=	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-28. Suspend - Short Data Sequence - Type 32 Single Slave

	BUS STATE					
SIGNAL LINES	Di	Di+1	Di+2	RD0	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	 DiH DiL S M	 Di+1H Di+1L S M	 Di+2H Di+2L S M	 RCD1 RCD0 S	 0 0	 0 AWS S
CYCLE TYPE Source=M	S	S	S	D	D	A
ACKNOWLEDGE Source=	RCG S	RCG S	ACK S	RCG S	RCG S	ACK S
WAIT Allowed Source=	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-29. Suspend - Extended Data Sequence - Type 16 Single Slave

	BUS STATE								
SIGNAL LINES	Di	Di+1	Di+2	RD0	RD1	::::	RD7	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	0 Di S M	0 Di+1 S M	0 Di+2 S M	0 RCD0 S	0 RCD1 S	:::: :::: S	0 RCD7 S	0 0	0 AWS S
CYCLE TYPE Source=M	S	S	S	D	D	::::	D	D	A
ACKNOWLEDGE Source=	RCG S	RCG S	ACK S	RCG S	RCG S	:::: S	RCG S	RCG S	ACK S
WAIT Allowed Source=	0 YES	0 YES	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES

Table 5-30. Suspend - Extended Data Sequence - Type 32 Single Slave

	BUS STATE								
SIGNAL LINES	Di	Di+1	Di+2	RD0	RD1	RD2	RD3	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	 DiH DiL S M	 Di+1H Di+1L S M	 Di+2H Di+2L S M	 RCD1 RCD0 S	 RCD3 RCD2 S	 RCD5 RCD4 S	 RCD7 RCD6 S	 0 0	 0 AWS S
CYCLE TYPE Source=M	S	S	S	D	D	D	D	D	A
ACKNOWLEDGE Source=	RCG S	RCG S	ACK S	RCG S	RCG S	RCG S	RCG S	RCG S	ACK S
WAIT Allowed Source=	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-31. Suspend - Type 16 Multiple Slave

	BUS STATE								
SIGNAL LINES	Di	Di+1	Di+2	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	0 Di M	0 Di+1 M	0 Di+2 M	0 0	0 AWM0 S	0 AWM1 S	0 AWM2 S	0 AWM3 S	0 AWM4 S
CYCLE TYPE Source=M	S	S	S	D	A	A	A	A	A
ACKNOWLEDGE	RCG	RCG	ACK	RCG	ACK	ACK (NS)	ACK (NS)	ACK (NS)	ACK (NS)
Source(7..0)=	S	S	S	S	S	S			
Source(15..8)=	S	S	S	S	S		S		
Source(23..16)=	S	S	S	S	S			S	
Source(31..24)=	S	S	S	S	S				S
WAIT Allowed	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-32. Suspend - Type 32 Multiple Slave

	BUS STATE								
SIGNAL LINES	Di	Di+1	Di+2	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	DiH DiL M	Di+1H Di+1L M	Di+2H Di+2L M	0 0	0 AWM0 S	0 AWM1 S	0 AWM2 S	0 AWM3 S	0 AWM4 S
CYCLE TYPE Source=M	S	S	S	D	A	A	A	A	A
ACKNOWLEDGE	RCG	RCG	ACK	RCG	ACK	ACK (NS)	ACK (NS)	ACK (NS)	ACK (NS)
Source(7..0)=	S	S	S	S	S	S			
Source(15..8)=	S	S	S	S	S		S		
Source(23..16)=	S	S	S	S	S			S	
Source(31..24)=	S	S	S	S	S				S
WAIT Allowed	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

5.3.5.2 Abort.

The Abort Sequence may be used to terminate a message prior to completion due to errors or other reasons. No header or acknowledge words are required for the Abort sequence. A master may initiate the Abort Sequence at any time during its tenure, except the last cycle of a Tenure Pass Message.

The Abort Sequence shall be as shown in Table 5-33. The first three of the four bus cycles with the AB cycle type are used to give the slave time to recognize that an abort has occurred. The slave (or slaves in the case of multicast or broadcast) shall respond by posting ACK on the AS lines during the third AB cycle to inform the master that an Abort has been recognized. There are no slaves on the fourth AB cycle and the master is the only module that may assert Wait on that cycle. After the last AB cycle, the master may continue with the H0 of another message or relinquish control of the bus by releasing the bus signal lines.

Table 5-33. Abort Sequence

	BUS STATE			
SIGNAL LINES	AB0	AB1	AB2	AB3
DATA D<31..16> D<15..0> Source=	 X X M or S	 X X M or S	 0 0	 0 0
CYCLE TYPE Source=M	AB	AB	AB	AB
ACKNOWLEDGE Source=	X X	X X	ACK S	NS
WAIT Allowed Source=	0 a a	0 a a	0 NO .	0 b b

a) Wait assertion allowed but not honored
b) Master Wait allowed, no slaves exist
X) not defined

5.3.6 WAIT.

A master or slave may assert Wait to control the data transfer rate on the PI-bus and thereby accommodate slow or temporarily busy devices. During a multiple slave sequence, one or more of the slaves may assert Wait. Each cycle on which Wait is asserted shall cause the insertion of a non-transfer cycle into a sequence.

5.3.6.1 Rules for Asserting Wait.

A Bus Interface may assert Wait on one or more cycles subject to the following restrictions:

1. Wait may normally be asserted by a Bus Interface only if that module is scheduled to be a bus master or slave on the next transfer cycle. However, the current bus master may assert Wait if the bus will be placed in the Idle state following the Wait induced non-transfer cycle(s) provided the vie interval requirement is not violated.

2. A Bus Interface shall not assert Wait on H0 or H1 cycles.

3. A Bus Interface shall not assert Wait on a particular cycle (N), if on the second previous cycle (N-2) the module did not assert Wait but Wait was asserted on that cycle. If the Bus Interface does assert Wait on cycle N and Wait was not asserted by any module on cycle N-1, the Bus Interface shall assert Wait for an even number of contiguous cycles.

5.3.6.2 Effects of Wait.

Other than during an Abort sequence, for each cycle that a Wait is asserted during a tenure, one non-transfer cycle (NT cycle) shall be inserted into the sequence immediately following the cycle in which Wait is asserted. The insertion of non-transfer cycles shall not change the sequence of scheduled bus states. However, if during the NT cycle, the master asserts an Abort sequence designation on the Cycle Type lines, the sequence shall be altered to Abort. During an Abort sequence, Wait shall not introduce non-transfer cycles.

5.3.6.2.1 Line Groups during Non-Transfer Cycles

During NT cycles produced by Wait, the signal line values shall be as specified below.

5.3.6.2.1.1 Data Line Group during NT Cycles. The value on the Data lines shall be ignored except for line error checking. Only valid symbols shall be posted on the Data line group.

The Data line group shall not be posted by any module other than the module which would have posted the Data lines had that cycle been the scheduled transfer cycle. If a module asserts a value on the Data line group on such an NT cycle, the value shall be a symbol which is valid for the next scheduled transfer cycle.

5.3.6.2.1.2 Cycle Type Group during NT Cycles. The value on the CT lines shall be ignored except for line error checking and the occurrence of Abort. An Abort Cycle Type shall mark the beginning of an Abort Sequence. The master module responsible for posting the CT lines on the next scheduled transfer cycle shall source a valid symbol on the CT lines during the NT cycle(s).

5.3.6.2.1.3 AS Group during NT Cycles. If a module is responsible for posting the AS lines on the next transfer cycle it shall post a valid symbol on the AS lines during the NT cycle(s). A symbol other than NAK shall be ignored during non-transfer cycles. NAK shall be posted on cycle N, to signal that an error associated with cycle N-2 was detected.

5.3.6.2.1.4 Wait Lines during NT Cycles. The rules for asserting Wait are specified in "5.3.6.1 Rules for Asserting Wait.." The effects of asserting Wait on a non-transfer cycle shall be the same as specified herein for asserting Wait on a transfer cycle.

5.3.6.2.1.5 Bus Request Lines during NT Cycles. The rules for asserting Bus Request are specified in "5.3.3.3 Bus Request.." Bus Request is completely independent of Wait.

5.3.7 DATA LINK FACILITIES.

This section specifies the Data Link facilities which shall be accessible over the PI-bus via the Bus Interface Message described in "5.3.4.4 Bus Interface Message Sequence."

5.3.7.1 Data Link Register Address Space.

The Data Link facilities specified herein shall be contained in the Data Link register address space which shall consist of 256 words assigned to consecutive addresses. This register space shall be allocated as shown in Table 5-34. Access to these facilities over the bus shall be allowed only via the Bus Interface Message with the Header Word A Access Type field set to zero (AT=000).

Reserved registers shall not be implemented and any attempt to access them shall cause a 'Resource Not Present' error. For write operations to defined registers, the state of any bits in the word which correspond to reserved bits shall be ignored.

Table 5-34. Data Link Register Address Space

ADDRESS	REGISTER
255 . . . 33	LOGICAL SLAVE IDENTIFICATION REGISTERS (33 - 255)
32 . . 6	RESERVED REGISTERS (6 - 32)
5	VIE PRIORITY REGISTER
4	VIE INTERVAL B REGISTER
3	VIE INTERVAL A REGISTER
2	MODULE CAPABILITIES REGISTER
1	CONTROL REGISTER
0	MULTICAST ACKNOWLEDGE REGISTER

5.3.7.2 Register Protection.

Access to the Data Link registers via the Bus Interface Message shall be limited by write protection. Either permanent or programmable write protection shall be provided for each register as specified in the following sections. The current state of programmable write protect must be controlled from the device. On reset, all programmable protection must be placed in the non-protect state.

5.3.7.3 Registers.

5.3.7.3.1 Multicast Acknowledge Register - Address 0.

The Multicast Acknowledge Register shall be a 16 bit register with the same format and field interpretations as the Single Slave Acknowledge word. During Multicast sequences, error information shall be accumulated by the slave. This information is identical to that required for the single slave case. The acknowledge word which would have been transferred to the master if the current sequence was a single slave sequence, shall be stored in the Multicast Acknowledge Register instead of being sent as an AWS, and the Multicast Acknowledge symbol shall be posted on the bus. Error indications shall apply to the current message only.

During multiple slave Bus Interface Message and Block Message sequences, a word equivalent to the Data Acknowledge word for a single slave sequence shall be formed. The acknowledge information stored in Multicast Acknowledge register bits <14..7> on the Header Acknowledge cycle shall be logically OR'ed with bits <14..7> of the equivalent single slave Data Acknowledge word and the result stored in bits <14..7> of the Multicast Acknowledge register on the slave's assigned Data Acknowledge cycle. Bit <15> and bits <6..0> of the equivalent single slave Data Acknowledge word shall be stored in Multicast Acknowledge Register bit <15> and bits <6..0> on the slave's assigned Data Acknowledge cycle. This Register shall be reset at the start of a multi-slave message in which this module is a participant.

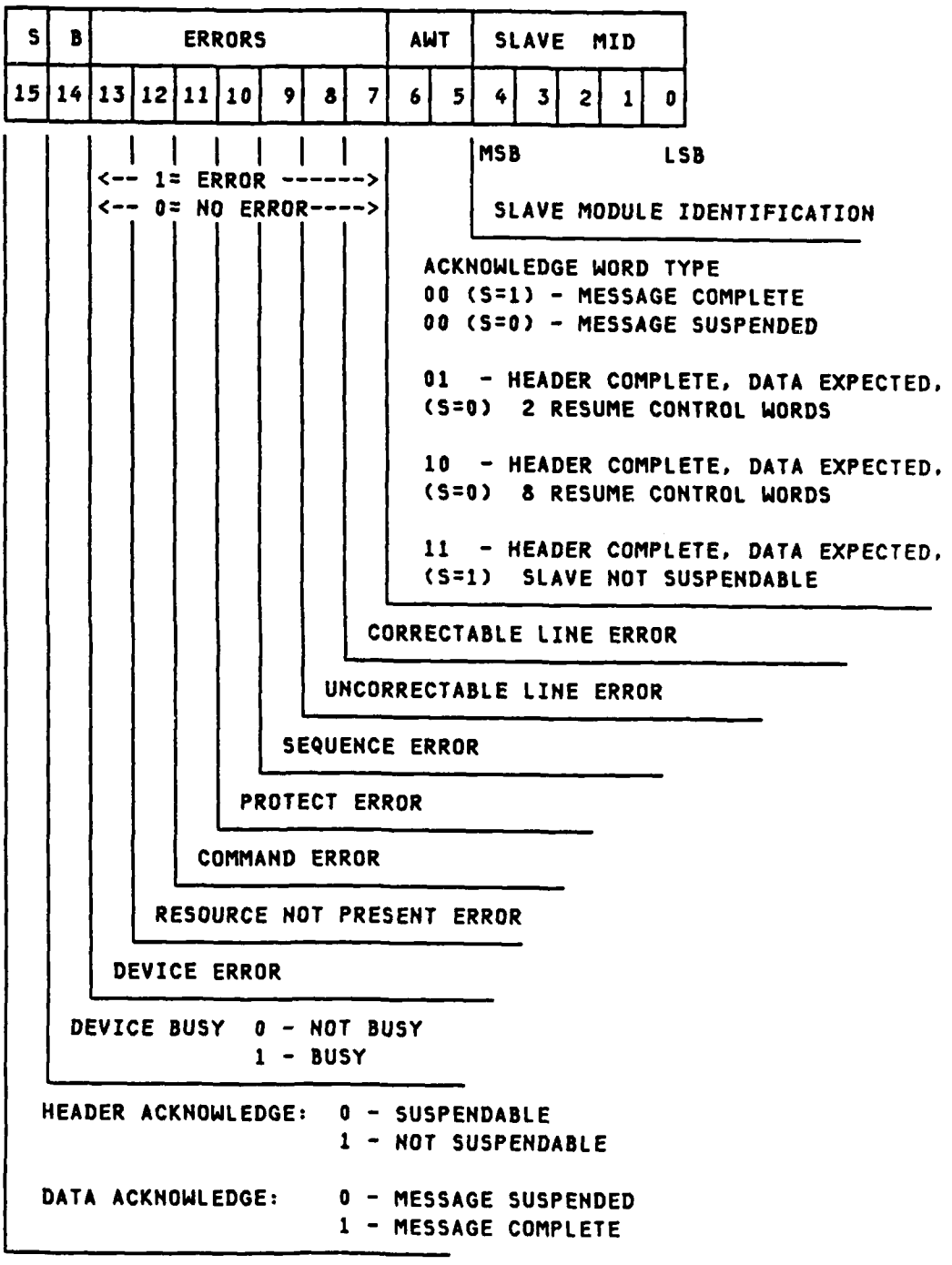
The following requirements shall apply to the Multicast Acknowledge Register:

Word format: Figure 5-11

Write protect: Permanent

State after reset: Bit <15>=1, Bits <14..5>=all zeros, Bits <4..0>=MID

Figure 5-11. Multicast Acknowledge Register Word Format



5.3.7.3.2 Control Register - Address 1.

The Control Register shall be a 2 bit register that shall contain a bit to initiate Bus Interface reset and a bit to initiate built-in-test.

The following requirements shall apply to the Control Register:

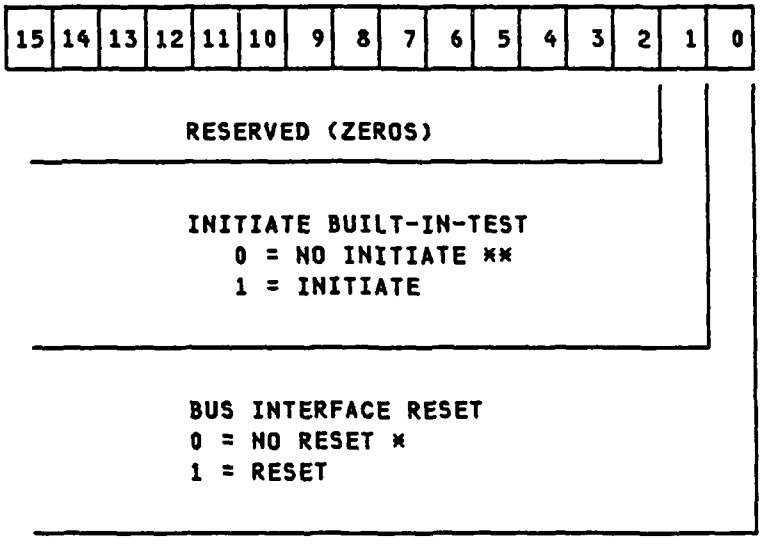
Word format: Figure 5-12

Write protect: programmable

State after reset: all zeros.

Reserved bits: read as zeros.

Figure 5-12. Control Register Word Format



* - Placed in 0 state at completion of reset.
** - Placed in 0 state at completion of 'built-in-test'.

5.3.7.3.3 Module Capabilities Register - Address 2.

The Module Capabilities Register shall be a 3 bit register that shall specify the physically implemented capabilities of the Bus Interface / Device Combination.

The following requirements shall apply to the Module Capabilities register:

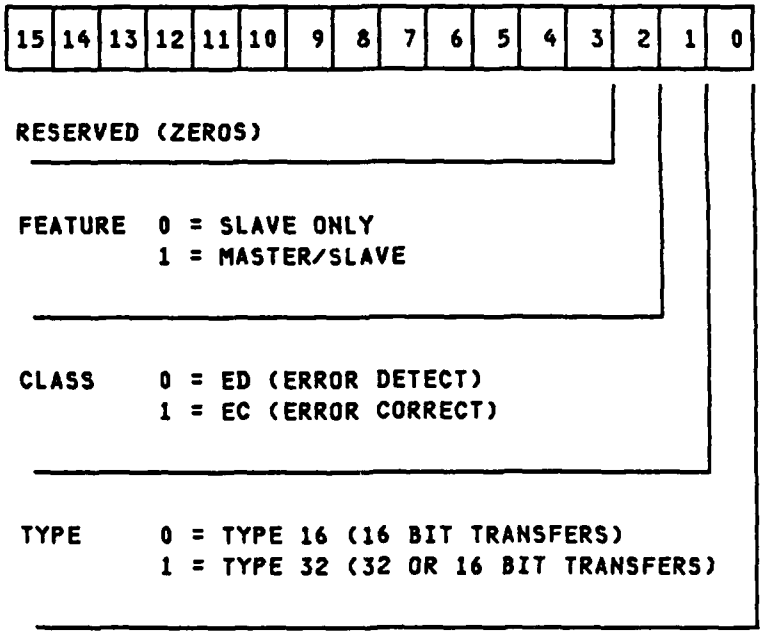
Word format: Figure 5-13

Write protect: permanent

State after reset: appropriate capabilities code

Reserved bits: read as zeros.

Figure 5-13. Module Capabilities Register Word Format



5.3.7.3.4 Vie Interval A Register - Address 3.

The Vie Interval A Register shall be a 16 bit register that shall contain the unsigned binary Vie Interval A timeout value expressed in bus cycles. The value in this register shall remain unchanged until a new timeout value is loaded. Reading this register shall return the last value written. This register is not required for Feature S0 modules. Any attempt to access a register which is not implemented shall cause a 'resource not present' error.

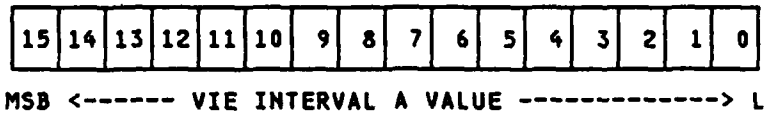
The following requirements shall apply to the Vie Interval A Register:

Word format: Figure 5-14

Write protect: programmable

State after reset: all ones

Figure 5-14. Vie Interval A Register Word Format



5.3.7.3.5 Vie Interval B Register - Address 4.

The Vie Interval B Register shall be an 8 bit register that shall contain the unsigned binary Vie Interval B timeout value expressed in bus cycles. The value in this register shall remain unchanged until a new timeout value is loaded. Reading this register shall return the last value written. This register is not required for Feature S0 modules. Any attempt to access a register which in not implemented shall cause a 'resource not present' error.

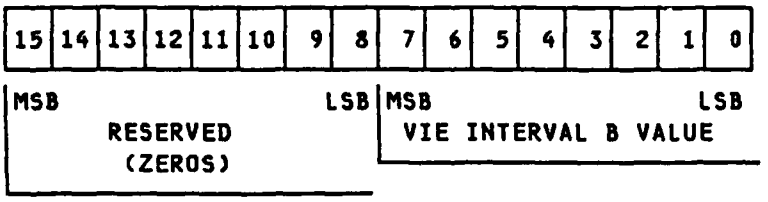
The following requirements shall apply to the Vie Interval B Register:

Word format: Figure 5-15

Write protect: programmable

State after reset: Bits <15..8>, all zeros; bits<7..0>, all ones

Figure 5-15. Vie Interval B Register Word Format



5.3.7.3.6 Vie Priority Register - Address 5.

The Vie Priority Register shall be a 12 bit register that shall contain the current vie priority of this module expressed in unsigned binary notation. The priority range is from zero (lowest priority), to 4095 (highest priority). This register is divided into two fields. The module identification field shall be fixed by the module's hardwired module identification code (MID) and cannot be changed by the Bus Interface Message. This register is not required for Feature S0 modules. Any attempt to access a register which is not implemented shall cause a 'resource not present' error.

The following requirements shall apply to the Vie Priority register:

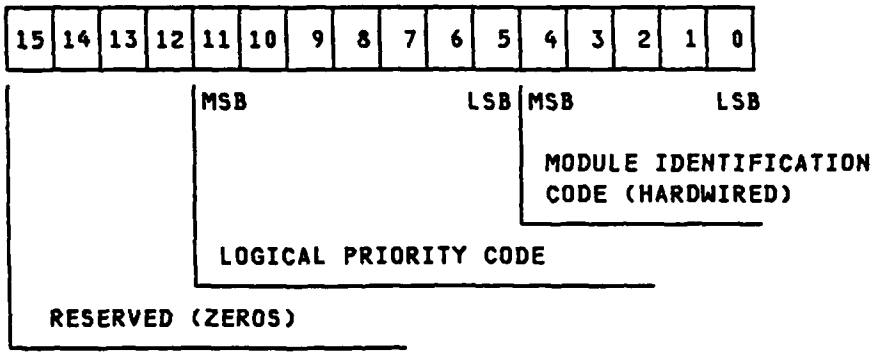
Word format: Figure 5-16

Write protect: Programmable for bits<11..5>; bits<4..0> fixed by MID

State after reset: Bits<11..5>, all zeros; bits<4..0>= MID bits<4..0>

Reserved bits: read as zeros.

Figure 5-16. Vie Priority Register Word Format



5.3.7.3.7 Reserved Registers - Addresses 6 To 32.

The use of the Reserved Registers shall be defined only by future versions of this specification. Until then, these registers shall not be implemented and shall cause a 'Resource Not Present' error if an access is attempted.

5.3.7.3.8 Logical Slave Identification Registers - Addresses 33 to 255.

The Logical Slave Identification Registers consist of a set of one bit registers in locations 33 to 255 of the Data Link register address space. The address of each register is identical to the Slave Identification code which the register controls. Bit 0 of each Logical Slave ID register indicates whether or not a Slave Identification code (ID) equivalent to the register address will be recognized by the module as a valid slave ID. The interpretation of bit 0 shall be:

1 = Bus Interface shall recognize this address as a slave ID.
0 = Bus Interface shall not recognize this address as a slave ID.

The following specifications shall apply to the Logical Slave Identification Registers:

Write protect: programmable

State after reset: zero

Reserved bits (15-1): read as zeros.

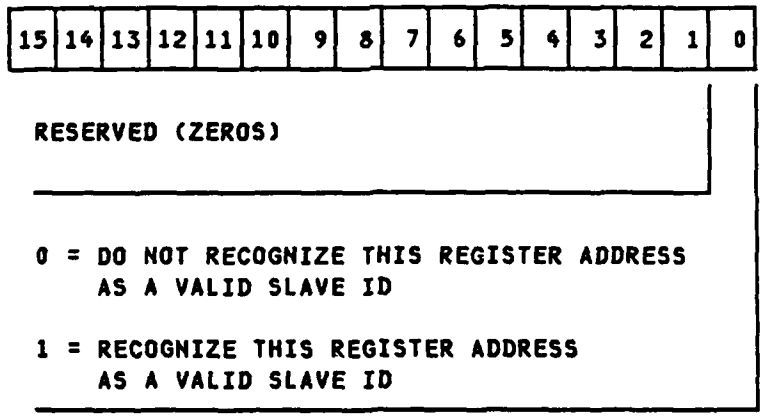
This register set is optional. A subset of the Logical Slave ID Registers may be implemented using either one or a combination of the two methods specified below:

1. The Logical Slave ID Registers may be implemented directly as a set of one bit registers in the Bus Interface register space. The slave identification registers shall have consecutively decreasing addresses starting at address 255. Any number of registers may be implemented from zero up to the full set of 223. The module shall not respond to sequences with Slave ID's corresponding to registers which are not implemented. Any attempt to access registers that are not implemented via a Bus Interface Message shall cause a 'resource not present' error.

2. The Logical Slave ID registers may be implemented indirectly using techniques such as associatively searching for valid slave ID's. In that case, the implementation shall specify the number of unique Logical Slave ID's which can be recognized (1 to 223) and each of those shall be capable of being set by the standard Bus Interface Message to any value in the range 33 to 255, inclusive. The standard Bus Interface Message shall also be capable of invalidating any previously validated slave identification code. Any attempt to validate more logical slave ID's than

allowed by the implementation shall cause a 'resource not present' error.

Figure 5-17. Logical Slave Identification Register Word Format



5.3.8 INITIALIZATION.

A reset mechanism shall be provided to initialize the Bus Interface. Reset shall be invoked by the attached device or by completion of a Bus Interface Message that set bit 0 (when not write protected) of the control register (see "5.3.7.3.2 Control Register - Address 1."). The Bus Interface shall respond to reset by becoming inactive, releasing all bus drivers and initializing Data Link registers to the values defined in "5.3.7 Data Link Facilities.." Following register initialization, modules may become active on the bus provided that MID<4..0>//MIP<0> satisfies P(MID<4..0>//MIP<0>) = 1. A module with incorrect MID parity, that is P(MID<4..0>//MIP<0>) = 0, shall not assert any PI-bus signal and shall not participate in PI-bus operations. A potential bus master module shall not initiate a Vie sequence prior to determining that the bus has been Idle for a minimum of two bus cycles and shall not assert Bus Request (BR) prior to determining that the current bus master is operating at a lower priority than the potential bus master's priority.

5.3.9 ERROR DETECTION, RECOVERY AND DIAGNOSTICS.

This section describes the detection of and responses to line, sequence and semantic errors. This section also defines PI-bus diagnostics requirements.

5.3.9.1 Correctable Line Errors

If the bus symbol error is correctable, then the symbol shall be functionally interpreted as the valid symbol with the least coding distance from the erroneous symbol. During message sequences, errors shall be logged in the Acknowledge word by setting the 'Correctable Line Error' bit to one (refer to "5.2.3.3.1 Single Slave Acknowledge."). The device should be notified of detected errors.

5.3.9.2 Uncorrectable Line Errors

If the bus symbol is uncorrectable, then the symbol shall be functionally interpreted according to Table 5-35. Any slave that detects an uncorrectable line error on the Data or CT lines shall post NAK on the second cycle after the cycle to which the error applies. During Vie, any module that detects an uncorrectable line error on the Data or CT lines shall post NAK on the second cycle after the cycle to which the error applies. The device should be notified of line errors. During message sequences, slaves shall log uncorrectable Data, Cycle Type, Acknowledge Set, Wait and Bus Request line errors in the Acknowledge word by setting the 'Uncorrectable Line Error' bit to one (refer to "5.2.3.3.1 Single Slave Acknowledge.").

Table 5-35. Interpretation and Response for Uncorrectable Invalid Symbols

SIGNAL LINE GROUP	INTERPRETATION AND RESPONSE
Wait	If Wait is not allowed: No Wait. If Wait is allowed: Wait asserted on first cycle of detected error and No Wait asserted on remaining contiguous error cycles.
Cycle Type	Scheduled Cycle Type (per defined sequences).
Cycle Type (during H0)	No slave ID match.
Bus Request	No Bus Request asserted.
Acknowledge Set	Assume NAK posted.
Acknowledge Set (during Vie)	If not the winner of vie, set master priority unknown. Should notify device.
Data lines (during Vie)	If a contender, assume that redundant bits which disagree are both asserted. If not a contender, set master priority unknown. If not the winner of vie, set master priority unknown.
Data lines (during Multi-slave acknowledgement)	For redundant bits which disagree, no acknowledgement for corresponding module.
Data lines (during H0)	No slave ID match.
Data lines (during header, except H0)	Slaves shall, discard header words, take no action based on header words, set the Acknowledge Word AWT field to 00, set the S field to 1 and become not selected after Header Acknowledge

5.3.9.3 Sequence Errors.

Each Bus Interface shall detect any error that causes a violation in the protocol syntax (sequence definitions). During message sequences, errors shall be logged in the Acknowledge word by setting the 'Sequence Error 'bit to one (refer to "5.2.3.3.1 Single Slave Acknowledge."). The device should be notified of detected errors.

5.3.9.3.1 Cycle Type Sequence Errors.

Every Bus Interface shall differentiate each bus cycle as belonging to one of the bus states listed in Table 5-8. Each slave in a sequence shall determine if the Cycle Type, as indicated by the CT lines, follows a legal sequence as defined under "5.2 GENERAL REQUIREMENTS.." The differentiation of legal and illegal cycle type sequences shall be by comparing the set of bus states that are defined as scheduled states for the current sequence to the actual sequence of symbols received on the CT lines. Table 5-36 shows the required response of a slave to received CT symbols (top row) verses scheduled bus states (left column). Definitions of the required responses are given in Table 5-37.

Table 5-36. Slave Response to Cycle Type Sequence Deviations

	CYCLE TYPE (CT) LINE SYMBOLS							
Scheduled Bus State	I	V	H0	H	D	A	S	AB
I	-	1	2	2	2	2	2	2
V0	2	-	error	error	error	error	error	error
V1, V2, V3	error	-	error	error	error	error	error	error
VZ0 .. VZ3	error	-	error	error	error	error	error	error
H0	4	error	-	error	error	error	error	5
H1 .. H9, HZ	error	error	error	-	error	error	error	5
HA0 .. HA4	error	error	error	error	error	-	error	5
HAZ	error	error	error	error	error	-	error	error
D	error	error	error	error	-	error	3	5
DZ	error	error	error	error	-	error	error	5
DA0 .. DA4	error	error	error	error	error	-	error	5
S0 .. S2	error	error	error	error	error	error	-	5
AB0 .. AB3	error	error	error	error	error	error	error	-

Table 5-37. Cycle Type Deviation Response Definitions

-	Proceed with normal operation.
error	Respond to Cycle Type sequence error as defined in "5.3.9.3.1 Cycle Type Sequence Errors"
1	Monitor vie process (too late to contend).
2	Ignore CT symbol, expect I cycle
3	Suspend message; refer to "5.3.5.1 Suspend.."
4	Expect I cycle (End-Of-Tenure). Set master priority = unknown.
5	Abort message; refer to "5.3.5.2 Abort.."

During Vie, a Bus Interface that detects an illegal CT sequence shall post NAK on the AS lines two cycles after each occurrence of the illegal Cycle Type. Modules which do not win the Vie sequence shall set master priority unknown as described in "5.3.3.1 Vie Sequence.."

If a Cycle Type other than H0 is received as the next Cycle Type following completion of a message sequence, Vie or Abort; each module shall assume that no HWA has been transmitted and, therefore, there is no match for that module's slave ID.

A Bus Interface that is a Slave in a sequence and has detected that the Cycle Type symbols have not followed a legal sequence shall post NAK for every such occurrence. The NAK shall be posted on the second cycle after the cycle that deviated from the legal sequence. If the illegal sequence leads to a condition in which the module cannot determine, with certainty, that the module should be a Slave, the Bus Interface shall not drive any line, other than Bus Request, when legal and required, until a valid H0 or Idle cycle is detected.

5.3.9.3.2 Acknowledge Set, Wait and Bus Request Sequence Errors.

Bus Interfaces shall detect Acknowledge Set, Wait and Bus Request sequence errors. Responses to these errors shall be as defined in Table 5-38. Modules shall not post NAK on the AS lines in response to AS, Wait nor Bus Request sequence errors.

Table 5-38. Sequence Error Response

SIGNAL LINE GROUP	SEQUENCE ERROR RESPONSE
Acknowledge Set	During vie, if not the winner set master priority unknown.
Wait	Assume Wait is not asserted.
Bus Request	Assume Bus Request is not asserted.

5.3.9.4 Semantic Errors.

5.3.9.4.1 Header Semantic Errors.

Slave modules shall respond to a reserved Message Type code in Header Word A by asserting NAK on the Acknowledge Set lines on the second cycle following H0 and becoming not selected as slaves on the following cycle. Slaves may also become not selected in response to specific slave device defined conditions.

Each Bus Interface shall detect any error in information transfer that has protocol significance. Table 5-39 lists and defines the errors in this category. In response to these errors, the Bus Interface shall post NAK on the AS lines within two cycles after the error is detected and not later than two cycles after the end of the message or partial message. The Bus Interface shall also log the error in the Acknowledge Word by making the bit named in Table 5-39 a logic 1. The device should be notified that an error was detected.

Table 5-39. Semantic Errors

SEMANTIC ERROR	ERROR DEFINITION
Protect ('Protect Error' bit)	A Write operation has been attempted on a write protected Bus Interface register.
Command ('Command Error' bit)	Header Word A has been received with a reserved code in the AT field. OR Header Word A has been received with a broadcast slave ID and a single slave Message Type. OR Header Word A has been received and the master's priority is unknown. OR A Tenure Pass Message Header Word A has been received with bits <7..5> asserted or AT not equal to 000 or F asserted or Header Word B bits <4..0> are not equal to the MID in HWA. OR A Type 16 module has been selected and the F bit in HWA is equal to 1. OR A Bus Interface Message Header Word A has been received with F asserted. OR A message has been received which is not in agreement with the defined format for the slave device.
Resource not Present ('Resource Not Present Error' bit)	A resource or capability has been addressed that is not implemented. OR A Tenure Pass Message has selected a Feature SO module as the slave. OR A non-existent or reserved Bus Interface register has been addressed.
Device ('Device Error' bit)	Module's device has detected an error attempting to perform bus related operation during the current message.

5.3.9.4.2 Header And Data Acknowledge Semantic Errors.

A bus master Bus Interface shall report semantic errors in the Header and Data Acknowledge Word to the device. The semantic errors which shall be reported are

1. the Acknowledge Word slave MID does not match the physical slave ID transmitted in Header Word A,

2. the Acknowledge Word Type (AWT) and/or Suspend (S) fields of the Acknowledge Word are not valid for the current Acknowledge cycle,

3. a Class ED master module receives an Acknowledge word with the "Correctable Line Error" bit set to logic 1 and

4. an Acknowledge word which has the "Protect Error" bit set to logic 1 is received when the MSG Type specified in HWA was not Bus Interface Message.

5.3.9.5 Diagnostics.

5.3.9.5.1 On-Line Testing.

On-Line testing of the PI-bus shall be accomplished through the Error Detection and/or Error Correction capability provided by the standard message sequences and protocol defined in previous sections of this specification.

5.3.9.5.2 Off-Line Testing.

Each Bus Interface shall be capable of transmitting and receiving arbitrary bit patterns on all signal lines of the bus (D, DC, CT, CTC, AS, W and BR) in parallel. Multiple clock cycles may be used to establish and read each pattern. Control and coordination of this test shall be through an alternate path independent from the PI-bus under test. The mechanism that coordinates the test should:

1. provide the line patterns to the Bus Interface transmitting the test pattern,

2. determine when the transmit pattern is stable,

3. read the received patterns from the receiving Bus Interface(s) and

4. analyze the patterns for correctness.

The mechanism that controls the test should apply patterns that guarantee detection of 1) a failed line stuck at zero or one and 2) a short between any two lines for any path from the signal line latch of the transmitting Bus Interface to the signal line latch of the receiving Bus Interface(s).

The transmitting Bus Interface shall be capable of being placed in the

off-line test mode, accepting patterns for transmission and transmitting the patterns. Once established, test patterns shall not be changed until a change is commanded by mechanism that coordinates the test.

The receiving Bus Interface, when in the off-line test mode, shall be capable of being commanded to clock in the test pattern from the bus and transfer that received pattern to the controlling device.

VHSIC Phase 2 INTEROPERABILITY STANDARDS

Appendix C

TM-Bus SPECIFICATION

November 9, 1987

Version 3.0

IBM

Honeywell

TRW

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

LIST OF ILLUSTRATIONS

1 SCOPE

1.1 Scope. This specification establishes the electrical, functional and performance requirements for the set of signal lines that constitute the Test and Maintenance Bus (TM-Bus).

1.2 Purpose. The purpose of this standard is to establish requirements for the TM-Bus and facilitate interoperability of modules which use the TM-Bus.

1.3 Intended Application. The TM-Bus is intended as a serial path for test and maintenance control and data information.

2 APPLICABLE DOCUMENTS

2.1 Government Documents. The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- VHSIC Phase 2 INTEROPERABILITY STANDARDS ETM-Bus SPECIFICATION, Version 3.0 dated November 9, 1986.
- VHSIC Phase 2 INTEROPERABILITY STANDARDS PI-Bus SPECIFICATION, Version 2.1 dated September 25, 1986.

2.2 Non-Government Documents. The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- None.

3 DEFINITIONS

The definitions listed here shall apply to the TM-bus and TM-bus modules.

3.1 Item Definition. The TM-bus is a linear, multi-drop communications medium which transfers bit serial data between a 'MASTER' module and up to 32 'SLAVE' modules residing on a single backplane. TM-bus modules implement the TM-bus protocol and meet all requirements of this specification.

Figure 1, illustrates the TM-bus and TM-bus modules. Conceptually, each module consists of a device which performs the application specific function of the module and a bus interface which implements the TM-bus MASTER-SLAVE communications protocol.

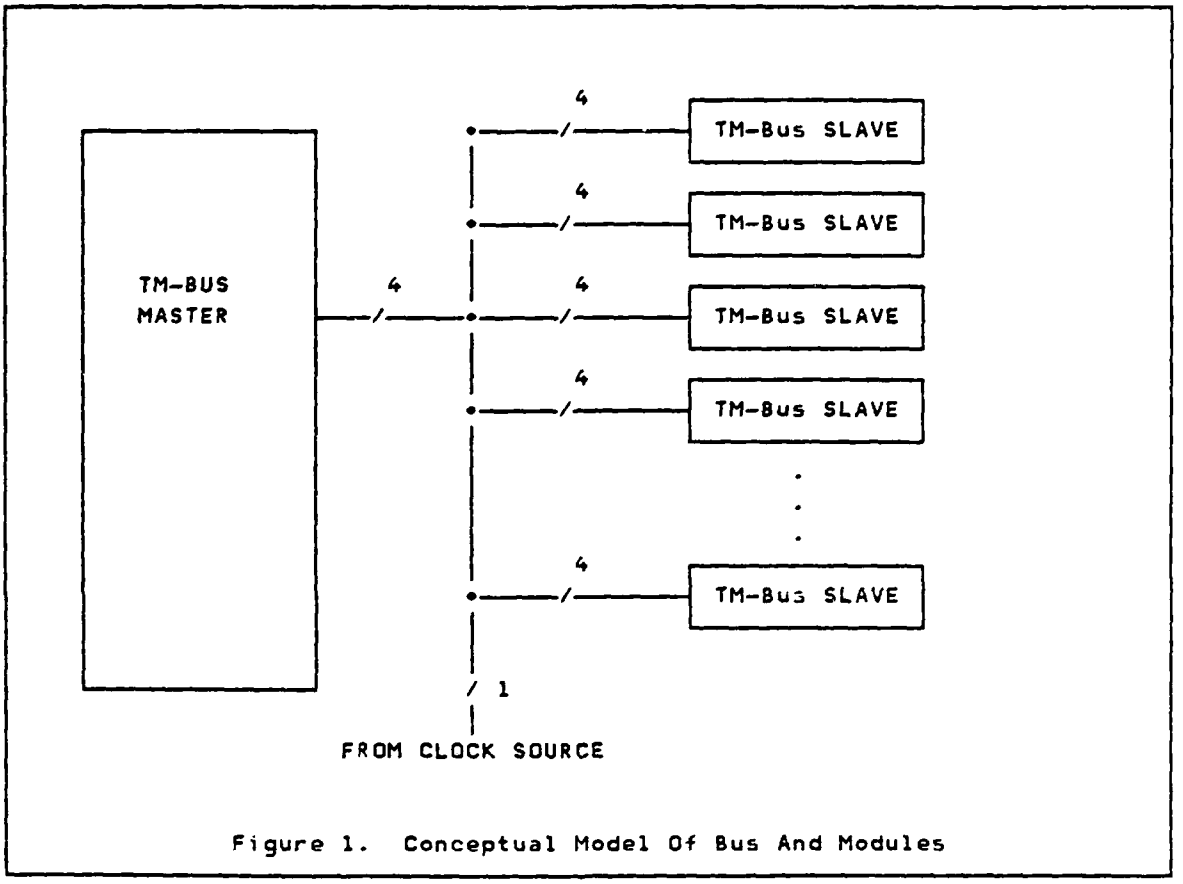


Figure 1. Conceptual Model Of Bus And Modules

3.2 Term Definitions.

active bus interface — A bus interface that is connected to the bus, and is currently capable of (and not inhibited from) participating in bus transactions.

assert — The action of changing the state of a bus signal line from released, logic 0, to asserted, logic 1, or of ensuring that the

line remains in the asserted state.

asserted	The logic 1 state of a bus signal line. The least positive of the two states of a bus signal line.
backplane	A motherboard comprising wiring for the bus and connectors to the modules attached to the bus.
broadcast	A mode of operation where the bus MASTER transmits data to all SLAVE modules during a single sequence.
bus MASTER	The module in control of the bus.
contend	When a bus SLAVE module(s) is/are actively vying for the attention of the bus MASTER.
device	The portion of a module, excluding the bus interface, which does the application dependent function of the module.
header	A sequence identifying a bus command, the SLAVEs participating in any commanded sequence and additional information delimiting the scope of the command.
linear bus	A bus with a single shared medium segment.
message	A set of sequences starting with a header and terminating when all bus actions indicated by that header have been performed.
module	An entity which is addressable via the bus and has a single bus connection.
module address	A pointer which uniquely identifies a module.
multicast	A mode of operation where the MASTER may transmit data to more than one SLAVE during a single sequence.
packet	A unit of data which is 17 bits, a 16 bit word plus 1 parity bit.
release	The action of ceasing to assert a logic 1 on a bus signal line. The action of releasing a signal line produces a change in the

	state of the signal line only if no module is asserting that signal.
released	The logic 0 state of a bus signal line produced when no module asserts the signal associated with that line. The more positive of the two states of a bus signal line relative to the 0 Volt logic reference.
response	A set of sequences sent by the SLAVE as a result of a message being sent by the MASTER.
sequence	An indivisible transaction comprising a number of transfers performing one intended function.
SLAVE	A module which does not have control of the bus but which is selected by the MASTER to participate in a sequence.
sub-address	A pointer to elements within an addressable module.
transfer	A set of elemental operations on the bus which result in the communication of bit serial datum units between the current bus MASTER and the selected SLAVE(s). A serial datum unit is 1 bit. See sequence.
word	An ordered set of 16 bits operated on as a unit. The most significant bit is labeled bit 16 and the least significant bit is labeled bit 1.

4 PHYSICAL LAYER

4.1 Introduction. The physical layer of the TM-Bus is specified herein. The lines required to implement the TM-Bus are defined, the electrical characteristics of the modules and backplane are specified and timing definitions are presented. The bus interface facilities which are accessible to a bus MASTER over the bus are also defined.

4.2 Line Definition. The TM-Bus signal, clock and module identification lines are defined in this section.

4.2.1 Nomenclature. Lines shall be designated by name. When a set of related bits are represented by the same name, the bits within the set shall be differentiated by number with the least significant bit numbered 0. All fields shall be referred to by their bit position within a data word transferred over the TM-bus. The nomenclature for single bits shall be the bit number enclosed in < >. The nomenclature <m..n> shall be the abbreviation for the set of bits m to n inclusive, where m and n are the most and least significant bits respectively.

4.2.2 TM-Bus Signal Definition. There shall be four (4) signal types that make up the TM-Bus as shown in Figure 2 on page 5. All bus signals shall use negative logic, i.e. the logic '1' state (or asserted state) is the lowest voltage level on the bus and the logic '0' (or released state) is the higher voltage level on the bus.

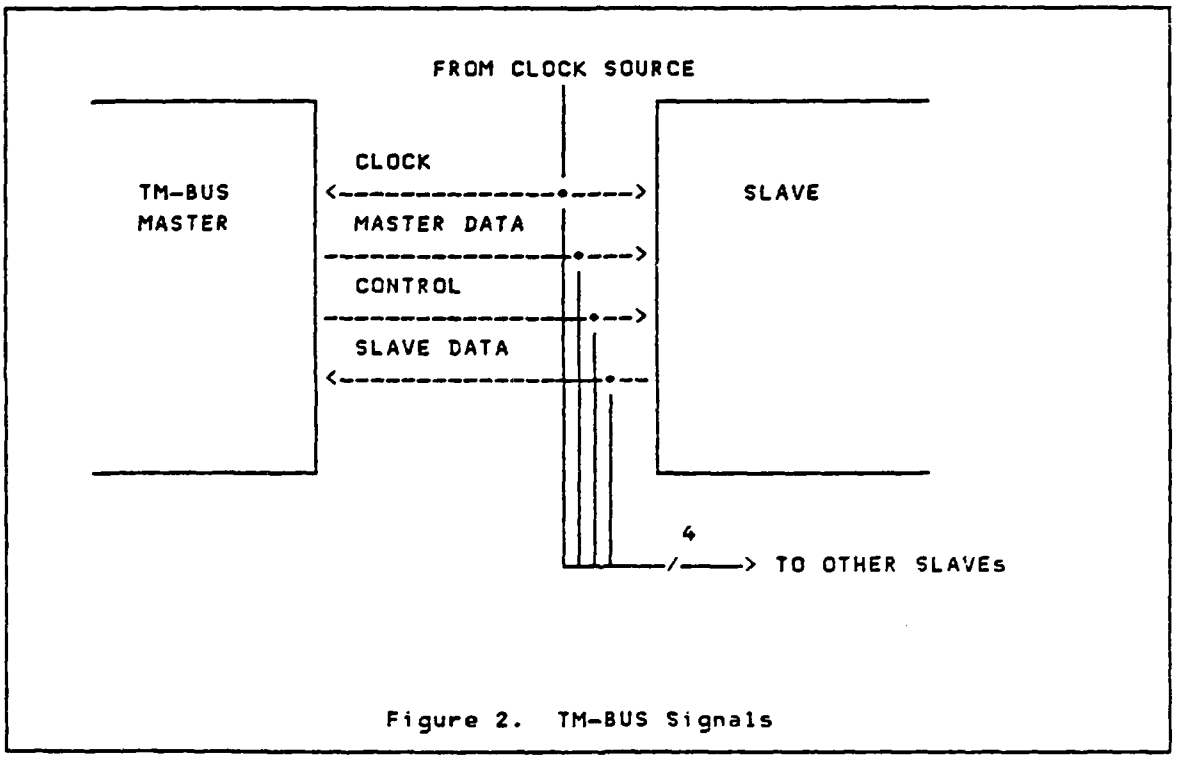


Figure 2. TM-BUS Signals

4.2.2.1 TM-Bus CLOCK Signal Definition. The TM-Bus CLOCK signal shall be a single phase clock. The TM-Bus interface shall support the full range of clock frequencies from zero (0) Hz to 6.25 MHz. All control and data transfer operations shall be synchronized with the TM-Bus CLOCK signal. All data and commands shall be placed on the TM-Bus on the high to low transition of the clock and latched-in on the next high to low transition.

4.2.2.2 TM-Bus MASTER DATA Signal Definition. The TM-Bus MASTER DATA signal shall be a single uni-directional line used to transmit device addresses, instruction data, and/or scan data from the MASTER to the SLAVE(s). The MASTER DATA line is also used in conjunction with the CONTROL line to indicate bus states (see Section "5.2.1 TM-Bus States" on page 13).

4.2.2.3 TM-Bus SLAVE DATA Signal Definition. The TM-Bus SLAVE DATA signal shall be used to transmit acknowledgements, data, and/or interrupts from the SLAVE(s) to the MASTER. The TM-Bus SLAVE DATA line shall support a wired-OR configuration.

4.2.2.4 TM-Bus CONTROL Signal Definition. The TM-Bus CONTROL signal shall be a single uni-directional line from the MASTER to the SLAVES(s). When the CONTROL line is asserted the bus is placed in the DATA TRANSFER state. When CONTROL is released, the bus is in the PAUSE or IDLE state.

4.2.2.5 TM-Bus Addressing. Each TM-Bus SLAVE is addressed by an eight bit address field. This address shall be sent in the HEADER packet containing the five (5) bit module address (bits <16..12>) and the three (3) bit sub-address (bits <11..9>), (see Figure 9 on page 15).

The five-bit module address field in the HEADER shall be compared to five Module IDentification (MID) inputs to determine if the SLAVE is being addressed. As a minimum, each SLAVE shall also have a Module Identification Parity (MIP) input that shall be set such that the modulo two sum of the five MID inputs and the MIP input equals one (1). (Note: the asserted state of each input is a logical one). When used in conjunction with the VHSIC Phase 2 PI-Bus, it is recommended that each TM-Bus SLAVE module have its MID and MIP inputs hardwired to the backplane of the TM-Bus (see section "4.2.4 Module Identification" of the PI-Bus Specification, Version 2.1 dated September 25, 1986). If an unrecoverable error occurs on the MID inputs, the TM-Bus SLAVE shall not execute any commands and shall release the SLAVE DATA line.

Comparison of the three (3) sub-address bits from the HEADER packet against Sub-address IDentification (SID) inputs is optional.

Module addresses '0' through '30' have a maximum of eight (8) subaddresses. Address '31' is limited to three (3) subaddresses ('F8', 'F9', and 'FA' HEX) due to restrictions of broadcast and multicast commands. See Section "5.2.5 Broadcast Capability" on page 22 and Section "5.2.6 Multicast Capability" on page 22 for details.

4.3 ELECTRICAL REQUIREMENTS. Electrical characteristics for the TM-bus backplane and modules shall be as specified herein. These requirements are the same as in Version 2.1 of the VHSIC Phase 2 PI-Bus Specification dated 9/25/86, Section "4.3 ELECTRICAL REQUIREMENTS".

4.3.1 Backplane Requirements

4.3.1.1 Bus Signal Line Characteristic Impedance. TM-Bus signal lines shall have a characteristic impedance of not less than 20 ohms and not more than 50 ohms for all operating and module loading conditions.

4.3.1.2 Bus Signal Line Termination. Signal lines shall be terminated at each end of the backplane to a circuit which is the Thevenin-equivalent of a terminating resistor in series with a voltage source of not less than +1.9 Volts nor more than +2.1 Volts. The value of the terminating resistance shall be between 30 and 40 ohms, inclusive.

4.3.1.3 Bus Signal Line Resistance. The series resistance for backplane signal lines shall be limited such that the maximum voltage rise from any asserted module output to the terminating resistance at either end of the backplane is less than 100 millivolts.

4.3.1.4 Module Identification Line Resistance. The resistance of the grounded MID and MIP lines with respect to the signal ground shall be less than 10 ohms.

4.3.1.5 Bus Clock Requirements

4.3.1.5.1 Voltage Levels. The low level voltage for Bus Clock shall be less than or equal to +0.55 volts. The high level voltage for Bus Clock shall be greater than or equal to +2.4 volts.

4.3.1.5.2 Rise And Fall Time. The rise time (Tr) of the Bus Clock from 0.8 volts to 2.0 volts shall be less than 5 nanoseconds. The fall time (Tf) of the Bus Clock from 2.0 volts to 0.8 volts shall be less than 5 nanoseconds.

4.3.1.5.3 Duty Cycle. The ratio of the Bus Clock high state duration to the bus clock period measured at 1.5 Volts shall not be less than 0.45 nor greater than 0.55.

4.3.2 Module Requirements

4.3.2.1 Bus Clock Requirements

4.3.2.1.1 DC Requirements

4.3.2.1.1.1 Input Capacitance. Bus Clock capacitance to logic ground shall be less than 22 picofarads.

4.3.2.1.1.2 Input Inductance. Bus Clock series inductance from the module input to the receiver of the signal shall be less than 27 nanohenries.

4.3.2.1.1.3 Bus Clock Current. The maximum current sourced by the module when the clock input voltage is +0.55 volts shall be 1.6 milliamps. The maximum current into the module when the Bus Clock voltage is +2.4 volts shall be less than 100 microamps.

4.3.2.1.1.4 High-level Input Voltage. An Bus Clock input voltage of +2.0 volts or more shall be interpreted as a high level.

4.3.2.1.1.5 Low-level Input Voltage. A Bus Clock input voltage of +0.8 volts or less shall be interpreted as a low level.

4.3.2.1.2 AC Requirements. Modules shall operate correctly with the Bus Clock characteristics specified in "4.3.1.5 Bus Clock Requirements."

The maximum Bus Clock frequency for the module shall be specified. The minimum Bus Clock frequency shall be zero Hertz.

All TM-bus timing shall be referenced to the high-to-low transition of Bus Clock through a voltage of 1.5 volts.

4.3.2.2 Signal Line Requirements

4.3.2.2.1 DC Requirements

4.3.2.2.1.1 Input Capacitance. Signal line capacitance to logic ground shall be less than 22 picofarads.

4.3.2.2.1.2 Input Inductance. Signal line series inductance from the module input to the driver or receiver of the signal shall be less than 27 nanohenries.

4.3.2.2.1.3 Leakage Current. Over the input voltage range of +0.3 volts to +2.1 volts, the absolute value of the output current for any signal line which is not being asserted by the module shall be less than 100 microamps.

4.3.2.2.1.4 Low-level Sink Current. The low-level output sink current (Iol) drive capability for signal lines shall be greater than 95 milliamps at an output voltage of 1.15 volts.

4.3.2.2.1.5 High-level Output Voltage. The high-level output voltage shall be determined by the backplane signal line termination voltage which is +1.9 to +2.1 volts. The signal line outputs shall permit wired-OR operations on the bus.

4.3.2.2.1.6 Low-level Output Voltage. The low-level output voltage (Vol) for signal lines shall be less than 1.15 volts at an input current of 95 milliamps.

4.3.2.2.1.7 High-level Input Voltage. A signal line input voltage (Vih) of +1.6 volts or more shall be interpreted as a logic 0. A signal line input which is not electrically connected to the backplane (i.e. an open line) shall be interpreted as a logic 0.

4.3.2.2.1.8 Low-level Input Voltage. A signal line input voltage (Vil) of +1.45 volts or less shall be interpreted as a logic 1.

4.3.2.2.2 AC Requirements

4.3.2.2.2.1 Signal Line Inputs. Figure 3 illustrates the timing relationships specified below.

4.3.2.2.2.1.1 Set-up Time. The maximum time that each input signal is required to be uniquely above or below the input voltage threshold for a logic 0 or logic 1 prior to the high-to-low transition of the clock (set-up time, Ts) shall be specified.

4.3.2.2.2.1.2 Hold-Time. The maximum time that each input signal is required to be uniquely above or below the input voltage threshold for a logic 0 or logic 1 following the high-to-low transition of the clock (hold time, Th) shall be specified and shall not exceed the minimum propagation delay time of the module.

4.3.2.2.2.1.3 Noise Rejection. The input signal lines shall reject and the Bus Interface shall not respond to any signal pulse whose width as measured between 1.5 volts on the low-to-high transition and 1.5 volts on the high-to-low transition is less than 4 nanoseconds.

4.3.2.2.2.2 Signal Line Outputs. The following specifications shall apply when the signal line is connected to the test circuit of Figure 4.

4.3.2.2.2.2.1 Propagation Delay. Propagation delay shall be measured with respect to the high-to-low transition of Bus Clock as illustrated in Figure 5. The reference clock voltage for timing shall be +1.5 volts. The reference signal voltage for timing shall be +1.5 volts.

The minimum and the maximum propagation delay (Tpdlh) for an output signal changing from a logic 1 (low voltage) to a logic 0 (high voltage) shall be specified for each output signal line.

The minimum and the maximum propagation delay (Tpdhl) for an output signal changing from a logic 0 (high voltage) to a logic 1 (low voltage) shall be specified for each output signal line.

4.3.2.2.2.2.2 Rise And Fall Time. The rise time (Tr) of an output signal from +1.2 volts to +1.8 volts shall be less than 9 nanoseconds. The fall time (Tf) of an output signal from +1.8 volts to +1.2 volts shall be less than 9 nanoseconds.

4.3.2.3 MID And MIP Lines. A binary 1 shall be represented by a connection to signal ground and a binary 0 shall be represented by an open circuit. Modules shall incorporate any circuits they require to sense the MID and MIP lines. The absolute current into a grounded MID or MIP line shall be less than 1 milliamp. The maximum voltage that shall exist on an open MID or MIP line shall not exceed 25 volts.

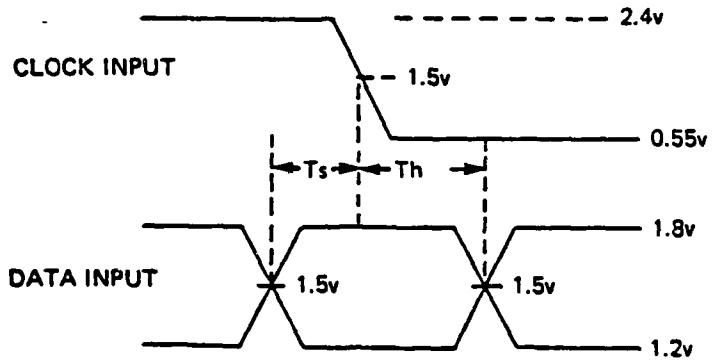


Figure 3. Set-up And Hold Timing

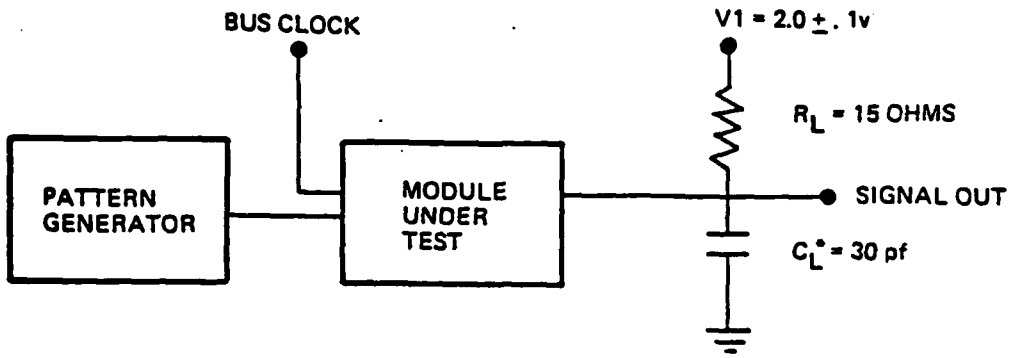


Figure 4. Signal Output Test Circuit

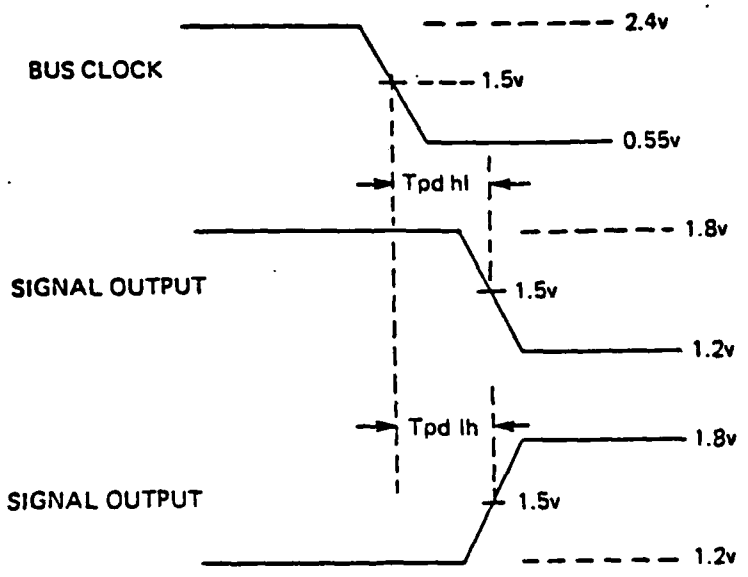


Figure 5. Output Signal Timing

5 DATA LINK LAYER

5.1 Introduction. The Test and Maintenance Bus (TM-Bus) shall be the channel for control and data information flow between the TM-Bus MASTER (e.g., a maintenance controller) and SLAVE modules within a system. The module in control of the TM-Bus shall be referred to as the MASTER and all other modules on the TM-Bus shall be referred to as SLAVEs. The information transferred and the scheduling of data and commands are system dependent and are not addressed in this specification. Figure 6 on page 12 summarizes the TM-Bus design parameters and characteristics.

o Performance Characteristics	o Protocol Characteristics
- 6.25 MHz clock (Typical)	- 8 reserved address bits
- 4 pin bus signals	- 32 module addresses (maximum)
- Synchronous Operation	- 8 sub-addresses per module address
- Two Data Lines	- Multi-drop Configuration
- SLAVE Status Register	- Interrupt Capability

Figure 6. TM-Bus Design Parameters and Characteristics

5.2 Operation. A message transmitted by the MASTER shall consist of a command HEADER packet, and optional DATA packets. If required, the SLAVE shall respond by acknowledging the HEADER and/or transmitting any DATA packets requested. The SLAVE shall only transmit packets when requested to do so by the MASTER. The SLAVE shall indicate interrupts as specified in Section "5.2.8 TM-Bus Interrupts" on page 24. All data shall be transmitted MSB first.

The following figures and paragraphs describe the operation of the four line serial TM-Bus in detail.

5.2.1 TM-Bus States. Using the Control and Master Data lines, the possible bus states are shown in Figure 7. Figure 8 on page 14 shows the state diagram for the TM-Bus.

CONTROL	MASTER DATA	STATE
0	0	IDLE/INTERRUPT (End of Message (EOM))
0	1	PAUSE/INTERRUPT
1	0	DATA TRANSFER/HEADER/CONTEND
1	1	DATA TRANSFER/HEADER/CONTEND

Figure 7. TM-BUS STATES

The IDLE state indicates that data shall not be transferred over the bus but interrupts from the SLAVEs are allowed over the SLAVE data line. The PAUSE state shall be used between packets during a message transfer to allow SLAVEs to interrupt the MASTER. The DATA TRANSFER state is entered from the HEADER or PAUSE state in the absence of a CONTEND command and indicates that data shall be transferred over the MASTER data line, the SLAVE data line, or both. The CONTEND sequence is entered from the HEADER or PAUSE state when the CONTEND command is issued.

Packets in a transmission may be separated by a variable number of PAUSE states (typically from 0 to 5, system requirements may dictate a higher number). The end of a message shall be signified by a return to the IDLE state.

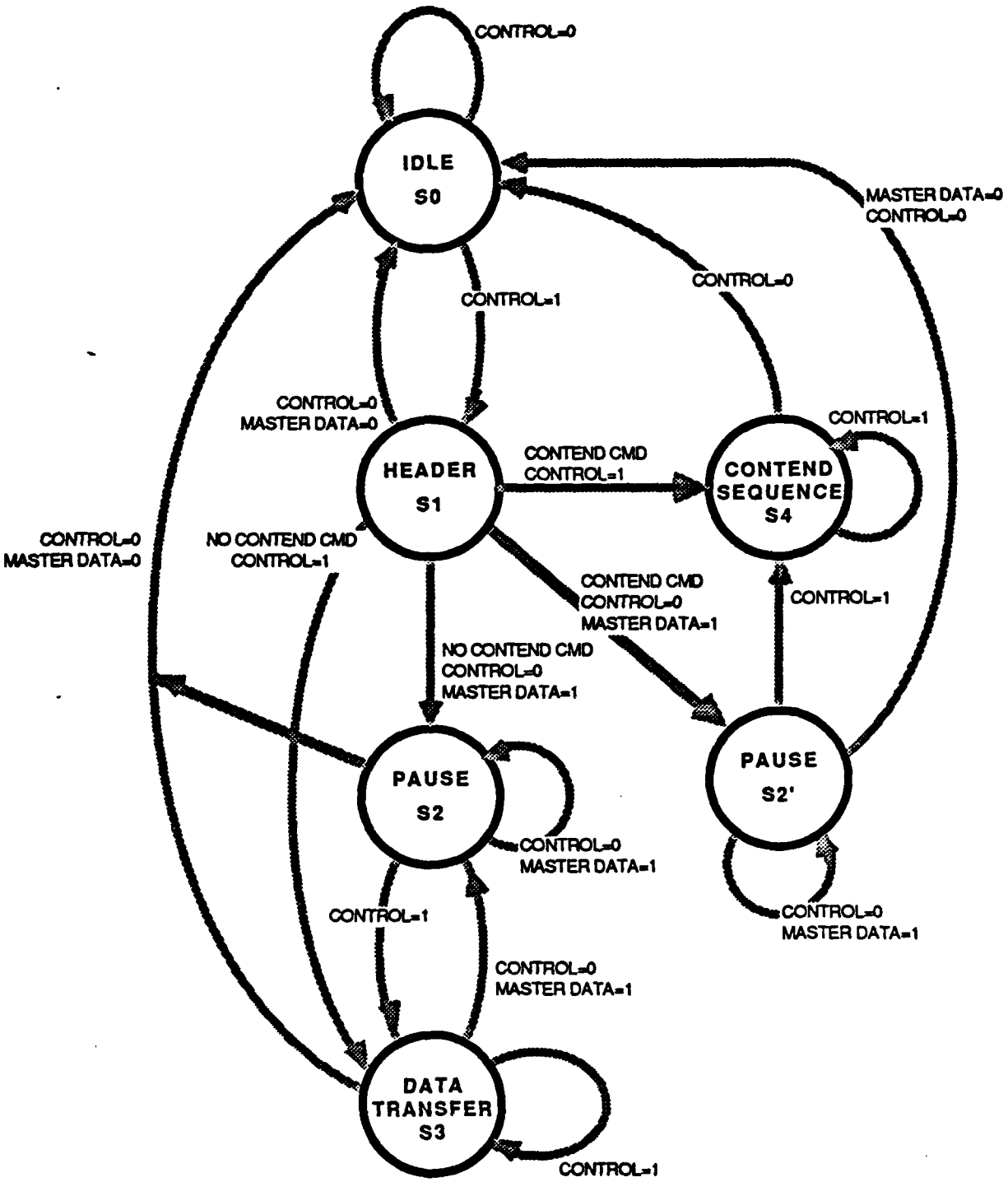


Figure 8. TM-Bus State Diagram

5.2.2 Message/Response Packet Descriptions. All messages sent by the MASTER shall consist of a HEADER and DATA packet(s). Responses (from the SLAVE) shall consist of an optional ACKNOWLEDGE packet and/or DATA packet(s). To allow flexibility, the number of DATA packets contained in a response is determined by user definable commands.

5.2.2.1 DATA PACKETS. The DATA packet contains sixteen (16) data bits (bits <16..1>) and one packet parity bit (bit <0>). The contents and format of the data bits are not specified. Data shall be sent MSB (bit 16) first.

5.2.2.2 Packet Parity. Bit 0 of each packet shall contain one packet parity bit. The parity shall be odd parity such that the modulo 2 sum of a data packet (bits <16..0>) = 1. The parity bit shall be transmitted last as the LSB.

5.2.2.3 HEADER Packets. Figure 9 shows the format for the HEADER packet which includes the SLAVE address and command fields. The SLAVE address field is eight (8) bits in length (bits <16..9>). The SLAVE command field is seven (7) bits in length (bits <8..2>). The ACKNOWLEDGE REQUEST field is one bit in length (Bit <1>).

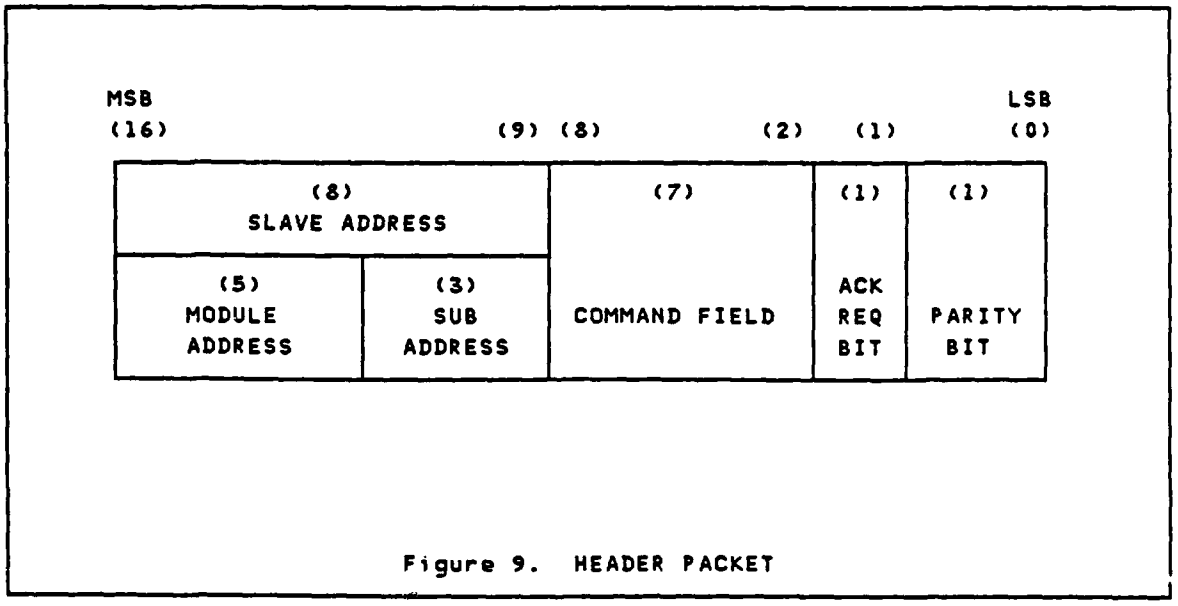


Figure 9. HEADER PACKET

The standard commands are defined in Section "5.3 Command Definitions" on page 27. If the ACKNOWLEDGE REQUEST bit (bit <1> of the HEADER) is asserted, the SLAVE shall respond with an ACKNOWLEDGE packet. Bit 0 is the packet parity bit.

5.2.2.4 ACKNOWLEDGE Packets. Figure 10 on page 16 shows the format for the ACKNOWLEDGE packet which includes the SLAVE address and status fields. The SLAVE address field is eight (8) bits in length (bits <16..9>) and contains the address of the responding SLAVE. The status field is also eight (8) bits in length (bits <8..1>) and contains the data residing in the SLAVE Status Register. Bit 0 is the parity bit.

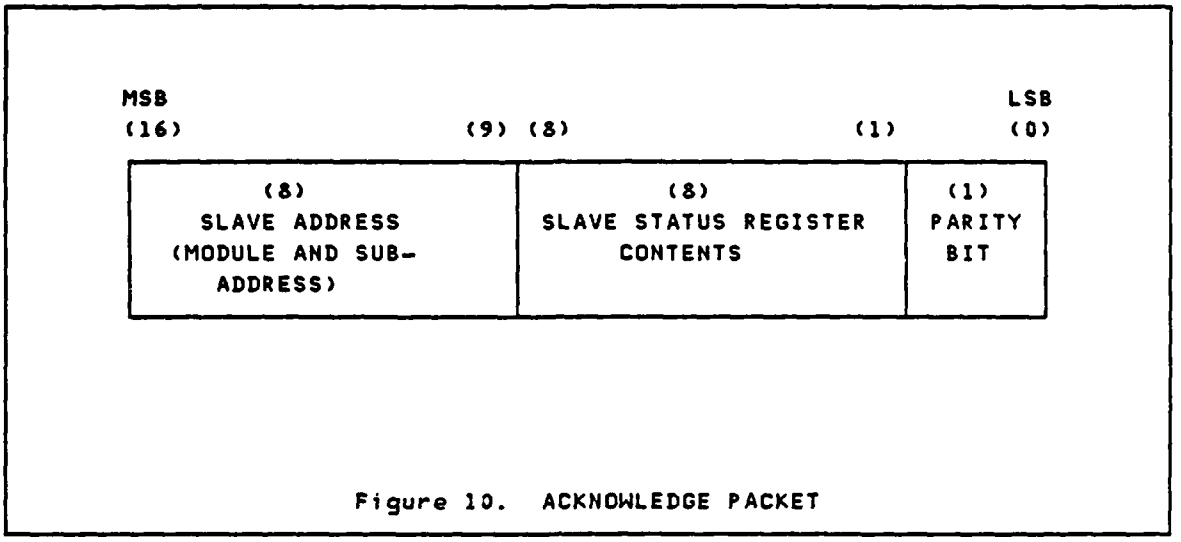
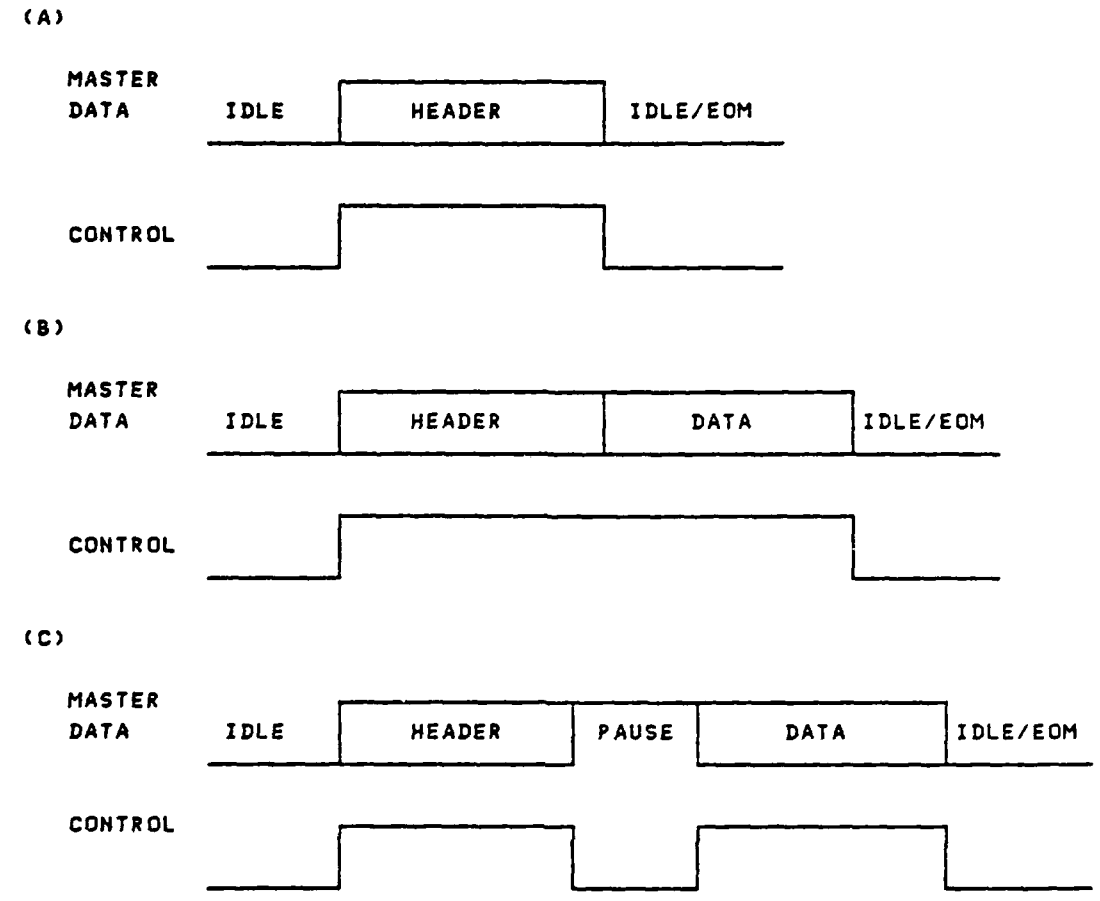


Figure 10. ACKNOWLEDGE PACKET

5.2.3 Message Protocol. A message transmission from the MASTER to a SLAVE shall be as shown in Figure 11. The MASTER shall begin a message transmission by first asserting the CONTROL line, which moves the bus from the IDLE state to the DATA TRANSFER state, and then transmitting the HEADER packet; the MASTER shall assert the CONTROL line for the duration of the packet transmission. At the end of a transmission, the MASTER shall release both the CONTROL and MASTER DATA lines, which returns the bus to the IDLE state. The MASTER shall move the bus to the PAUSE state between packets by releasing the CONTROL line while asserting the MASTER DATA line. If the message is longer than one packet, the MASTER shall assert the CONTROL line during each additional packet transmission. Optional PAUSE states are permitted between packets within a single message. The number of PAUSE states shall be determined by system requirements.



NOTE: The logic state is represented not the voltage level.

Figure 11. MESSAGE PROTOCOL

5.2.4 Response Protocol. All states on the SLAVE DATA line including PAUSE, IDLE, and packet transmissions, shall be synchronous to the MASTER DATA and CONTROL lines with a two clock cycle delay as shown in Figure 12 on page 18. This delay is required so that the SLAVE can receive and react to state transitions on the MASTER DATA and CONTROL lines. Following receipt of a HEADER packet, the addressed SLAVE shall begin packet transmissions on the SLAVE DATA line as soon as the CONTROL line is asserted (with the two cycle delay), as shown in Figure 13 on page 19 and Figure 14 on page 20. A flow diagram of the SLAVE response is shown in Figure 15 on page 21.

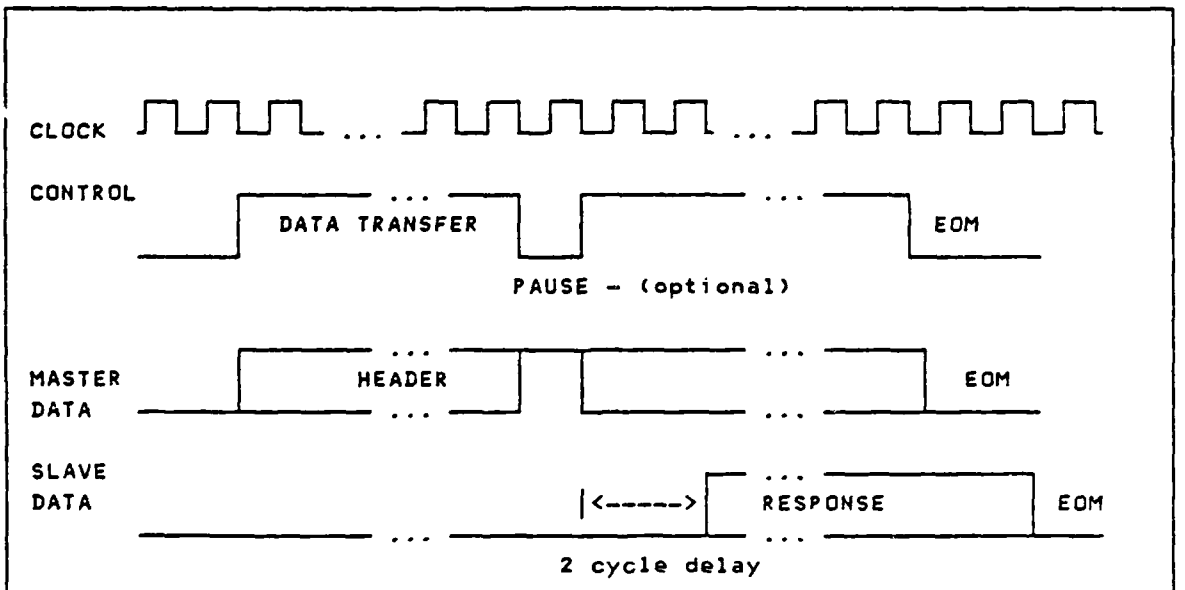
If the ACKNOWLEDGE bit is asserted in the HEADER, the addressed SLAVE shall respond with an ACKNOWLEDGE packet during the next packet transmission peri-

od. After the optional ACKNOWLEDGE packet, the addressed SLAVE shall transmit any DATA packets required by the decoded command, as shown in Figure 14 on page 20. The SLAVEs shall not respond with an acknowledge during broadcast or multicast operations.

If the message to the SLAVE contains DATA packets as shown in Figure 14 and the acknowledge bit is asserted, then the SLAVE shall respond by sending an ACKNOWLEDGE packet during the next period that data may be sent over the SLAVE DATA line. After the optional ACKNOWLEDGE packet, the SLAVE shall transmit any DATA packets required by the command.

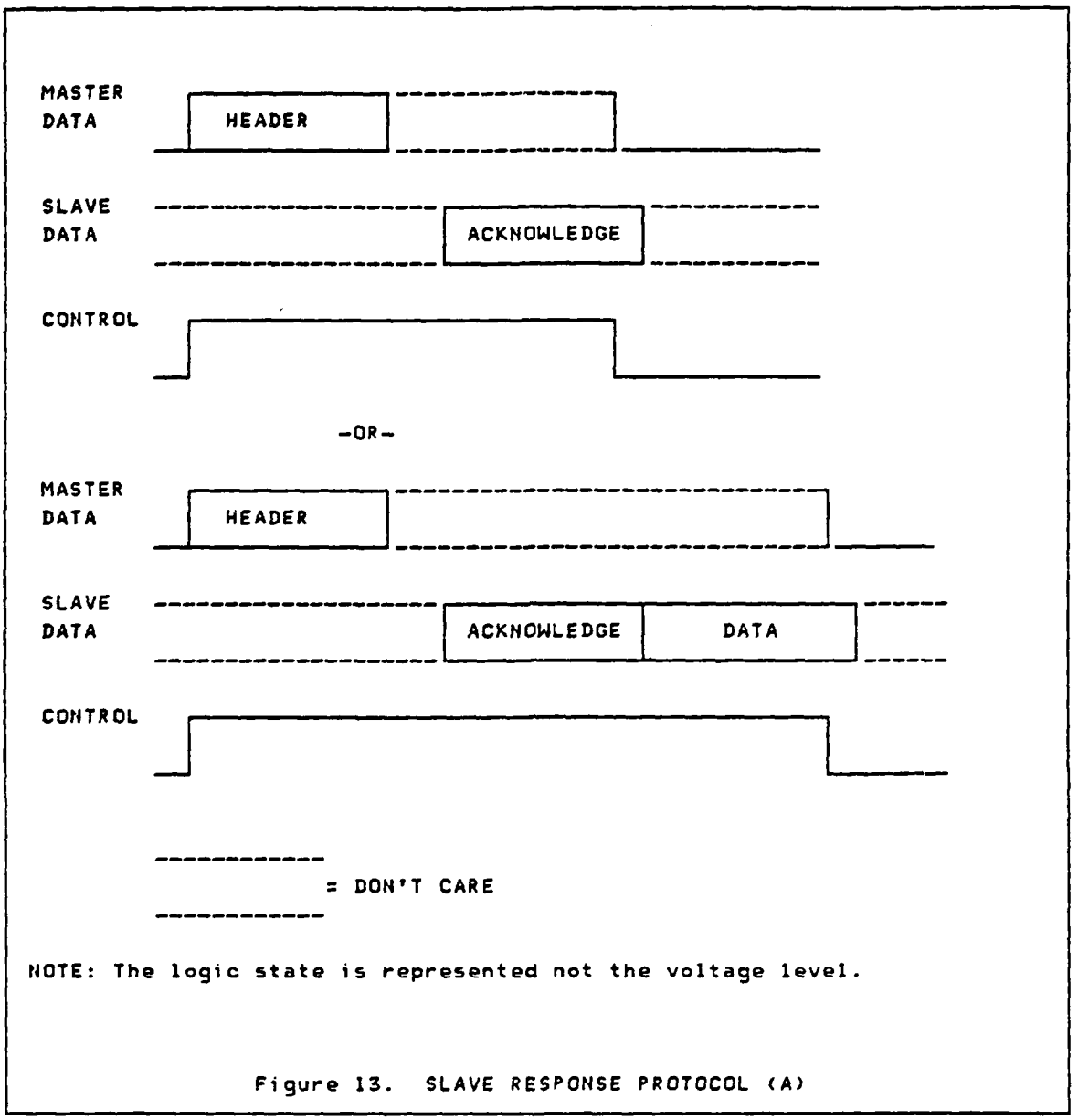
As shown in Figure 14 on page 20 data transfer may occur simultaneously on the MASTER DATA line and the SLAVE DATA line. This simultaneous transfer is dependent on the command received by the SLAVE. None of the standard commands shall require simultaneous transmissions.

Interrupt Handling protocol is described in Section "5.2.8 TM-Bus Interrupts" on page 24.



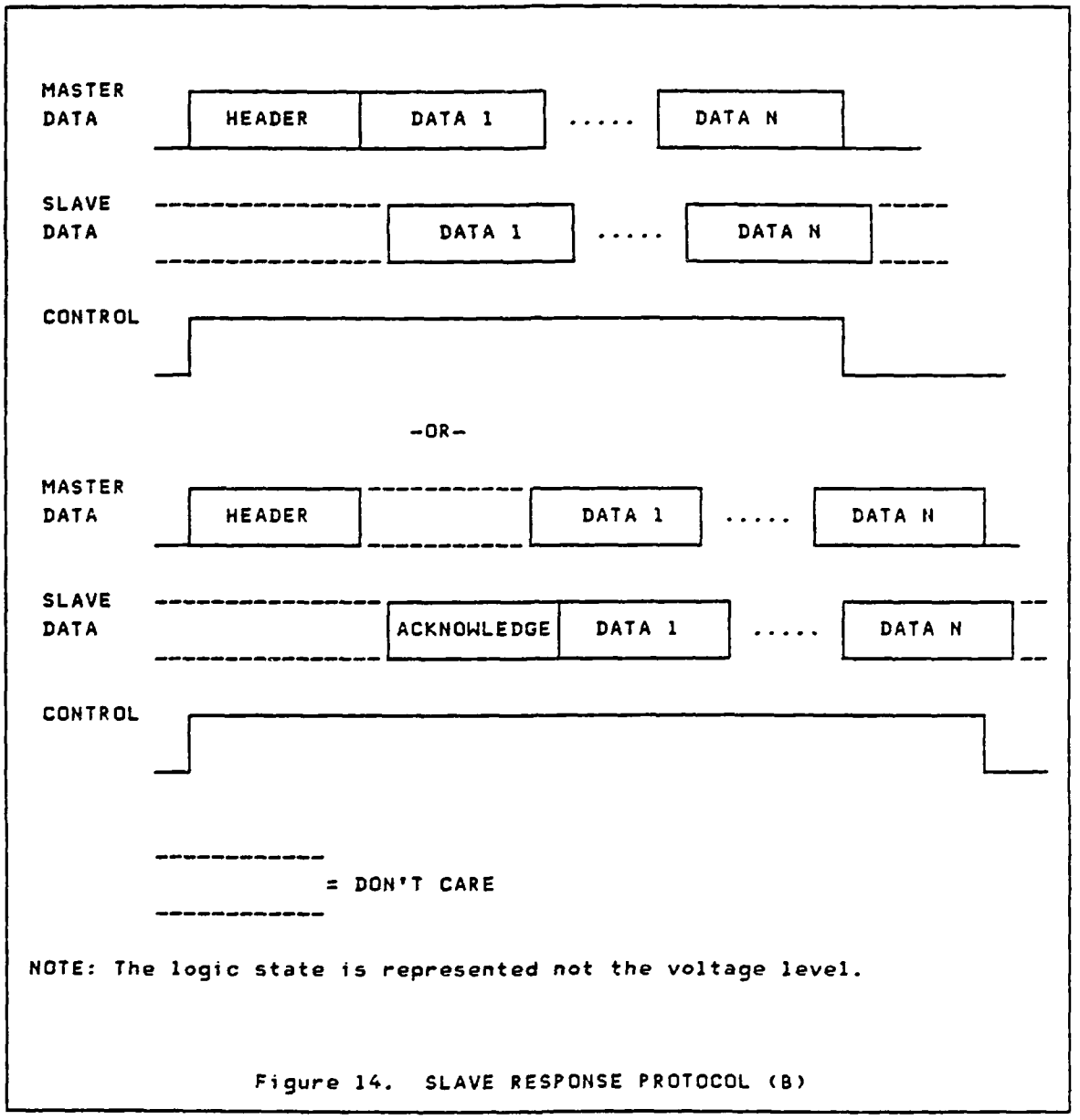
NOTE: The logic state is represented not the voltage level.

Figure 12. DATA TRANSFER TIMING



NOTE: The logic state is represented not the voltage level.

Figure 13. SLAVE RESPONSE PROTOCOL (A)



NOTE: The logic state is represented not the voltage level.

Figure 14. SLAVE RESPONSE PROTOCOL (B)

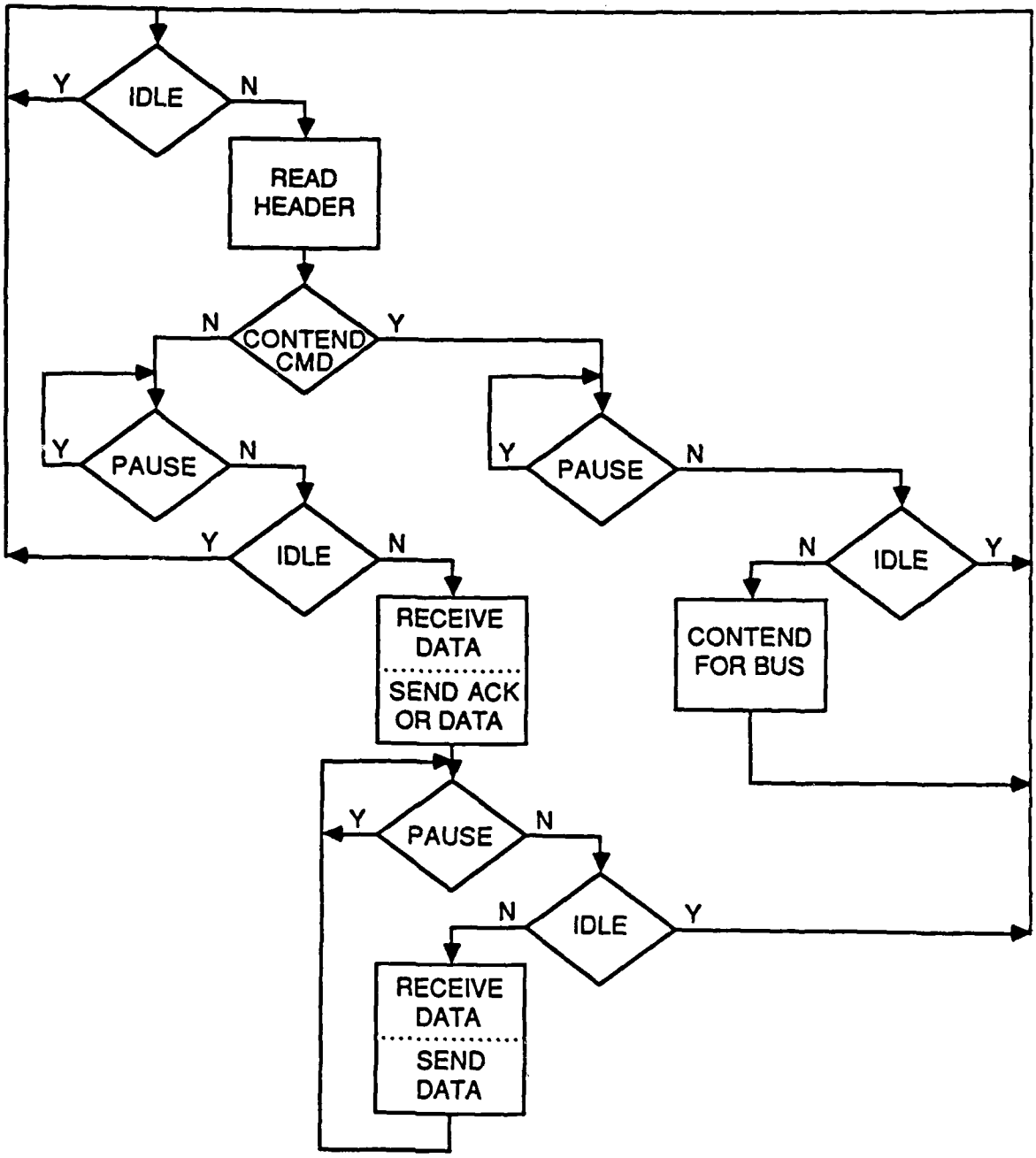


Figure 15. SLAVE RESPONSE FLOW

5.2.5 Broadcast Capability. The MASTER shall have the capability to broadcast to all SLAVEs by setting the SLAVE address field equal to ('FB' HEX). All SLAVEs shall recognize this address in addition to their normal module address and/or sub-address. SLAVEs shall indicate correct receipt of a broadcast command (HEADER packet) by asserting the Broadcast/Multicast Received bit in the SLAVE status register. The Broadcast/Multicast Received bit shall be released if the broadcast command was not received correctly or the SLAVE was busy during a broadcast operation, such that it could not execute the TM-Bus command. See sections "5.3.1 Reset SLAVE" on page 27, "5.3.3 Read Status Register" on page 27, and "5.6 TM-Bus Error Handling" on page 30 for further discussions of the broadcast/multicast received bit. Note that a SLAVE shall have the capability to execute the standard commands regardless of their busy state. A SLAVE shall assert the SLAVE Busy or the Bus Error bit in the SLAVE Status Register if a broadcast command is not received properly. SLAVEs shall not transmit any response packets over the SLAVE DATA line in response to broadcast operations except during CONTEND commands. SLAVEs may issue interrupts during a broadcast operation (see section "5.2.8 TM-Bus Interrupts" on page 24).

5.2.6 Multicast Capability. The MASTER shall have the capability to multicast to any number of SLAVEs. Each SLAVE shall belong to one of four multicast groups (00, 01, 10, 11), as indicated by the multicast select bits in the SLAVE Status Register (see Figure 16 on page 23). When a SLAVE receives a Reset SLAVE command, it shall set its group to '00'. A SLAVE shall change its group in response to each Multicast Select Command or Reset SLAVE command which it receives.

SLAVEs shall recognize four addresses as valid multicast addresses; ('FC' Hex), ('FD' HEX), ('FE' HEX) and ('FF' HEX) which shall be used for multicast groups '00', '01', '10', and '11' respectively. A SLAVE shall indicate correct receipt of a multicast command (HEADER packet) by asserting the Broadcast/Multicast Received bit in the SLAVE status register. The SLAVE shall release the Broadcast/Multicast Received bit if the multicast command was not received correctly or the SLAVE was busy during a multicast operation, such that it could not execute the TM-Bus command. See sections "5.3.1 Reset SLAVE" on page 27, "5.3.3 Read Status Register" on page 27, and "5.6 TM-Bus Error Handling" on page 30 for further discussions of the broadcast/multicast received bit. A SLAVE shall have the capability to execute the standard commands regardless of its busy state. A SLAVE shall assert the SLAVE Busy or the Bus Error bit in the SLAVE Status Register if it does not receive a multicast command properly. SLAVEs shall not transmit any packets over the SLAVE DATA line in response to multicast operations, except during Contend commands. See section "5.3 Command Definitions" on page 27 for details of Multicast Select commands. SLAVEs may issue interrupts during a multicast operation (see section "5.2.8 TM-Bus Interrupts" on page 24).

5.2.7 TM-Bus SLAVE Status Register. Each SLAVE shall have a SLAVE Status Register, described in Figure 16 on page 23. All bits in the status register shall be considered active when asserted. After returning the SLAVE status in response to a Read Status Register command, a SLAVE shall reset (release) the Bus Error, Broadcast/Multicast Received, and Event Occurrence bits (resetting the Reserved bit is optional). The SLAVE shall reset the Bus Error, Broadcast Received, and Event Occurrence bits when it wins a contend sequence (resetting the Reserved bit is optional).

Bit	Name	Meaning When Active
8(MSB)	Reserved	Available for user defined status. May be used for address extension.
7	SLAVE Busy	Indicates that the application side of the TM-Bus interface is busy.
6	Event Occurrence	Indicates that an error condition or other predefined condition exists.
5	Broadcast/ Multicast Received	Indicates that the last Broadcast/ Multicast command was properly received.
4	Bus Error	Indicates that a parity error or an illegal command has been detected by the SLAVE.
3	Multicast Select Bit 1	Indicates SLAVE multicast select Mode.
2	Multicast Select Bit 0	Indicates SLAVE multicast select Mode.
1(LSB)	Interrupt Enabled	Indicates whether the SLAVE may send an interrupt.

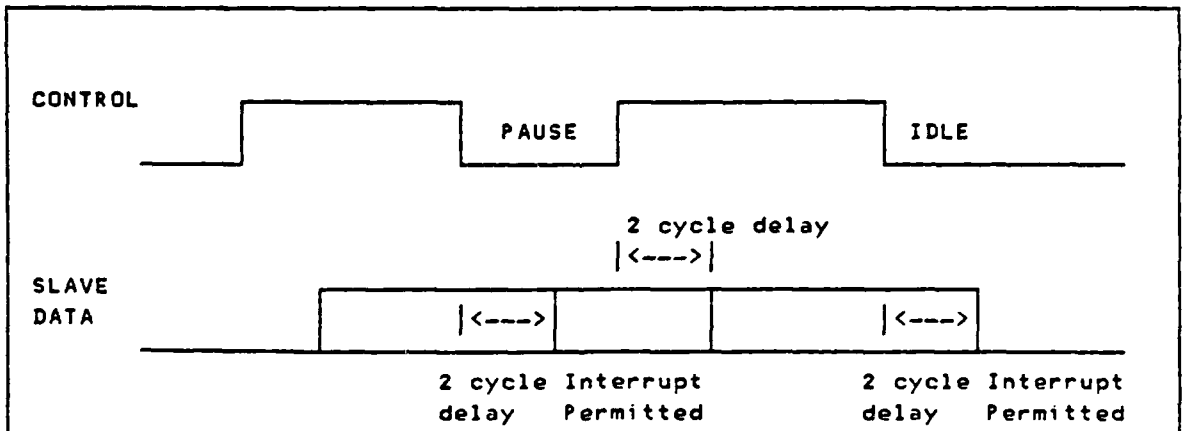
Figure 16. SLAVE Status Register

5.2.8 TM-Bus Interrupts. Any SLAVE may signal an interrupt to the MASTER by asserting the SLAVE DATA line for one clock period/cycle during the PAUSE or IDLE states, (when interrupts are enabled) as shown in Figure 17. On receiving an interrupt, the MASTER may service that interrupt by issuing a 'CONTEND for bus' command, checking the error status bits, and taking appropriate action.

The SLAVE shall send an interrupt out over the SLAVE DATA line when the Event Occurrence or Bus Error bits are asserted. The SLAVE shall consider the interrupt condition serviced when the SLAVE wins a contend sequence or the MASTER issues a Read Status Register command to that SLAVE. The SLAVE shall continue to send the interrupt for one clock period after all subsequent contend sequences, that the SLAVE does not win, until the interrupt is serviced. All interrupts shall be sent only during periods that interrupts are valid on the bus.

If the SLAVE Busy bit is asserted during data transfers (Bus State S3) after the optional ACKNOWLEGE packet is transferred, the SLAVE shall send an interrupt over the bus.

Any SLAVE that is currently addressed shall have the ability to interrupt during PAUSE states within a message. An active SLAVE's interrupt capability shall override the DISABLE INTERRUPT command. The SLAVE shall go back to the state selected by the last DISABLE or ENABLE INTERRUPT COMMAND following completion of a bus transaction.

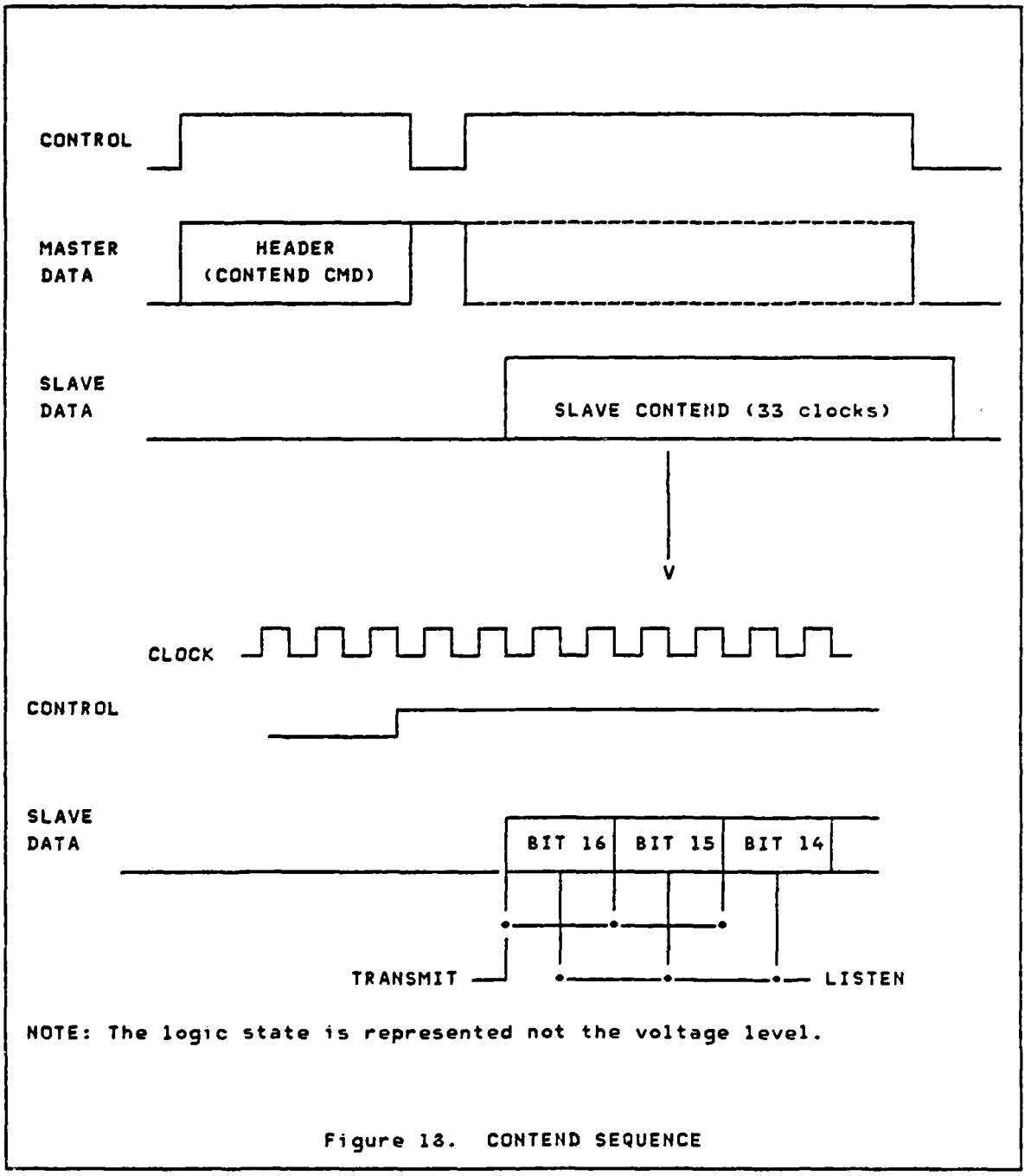


NOTE: The logic state is represented not the voltage level.

Figure 17. INTERRUPT TIMING

5.2.9 TM-Bus Contention. When the CONTEND for bus command is issued, any SLAVEs which meet the interrupt conditions described in Section "5.2.8 TM-Bus Interrupts" on page 24 (Event Occurrence bit asserted, Bus Error bit asserted, etc.) and for which interrupts are enabled may contend. During the transmission of this packet, the SLAVE shall 'listen' to the SLAVE DATA line and inhibit data transmission if a higher priority address is 'heard'. The highest SLAVE address shall have the highest priority.

To contend for the bus, a SLAVE sends its ACKNOWLEDGE packet, which includes the eight-bit SLAVE address. The SLAVE sends one SLAVE-address bit every two cycles, most significant bit first. After transmitting a bit during the first of the two cycles, the SLAVE 'listens' during the following cycle for a higher address; i.e., if the SLAVE has transmitted a '0', it listens for a '1'. If the higher address is not 'heard', then a SLAVE shall continue to alternately transmit and 'listen' until the entire 16 bit SLAVE ACKNOWLEDGE packet has been placed on the bus (in 32 clock cycles) and one bit of parity in the 33rd cycle, or until a higher address is 'heard' on the bus (as shown in Figure 18). After 33 cycles, the highest priority SLAVE that was contending has transmitted its ACKNOWLEDGE packet to the MASTER and the bus returns to the IDLE state.



NOTE: The logic state is represented not the voltage level.

Figure 18. CONTEND SEQUENCE

5.3 Command Definitions. The HEADER commands shown in Figure 19 on page 29 are defined below. There shall be 7 bits allowing 127 SLAVE commands. Commands 0 through 15 shall be standard or reserved commands and the remainder shall be user-defined commands. The command ('7F' HEX) with the ACKNOWLEDGE REQUEST bit asserted shall be an illegal command to all SLAVEs. If the '7F' command is detected by a SLAVE, with the ACKNOWLEDGE REQUEST bit asserted, then the SLAVE shall set the Bus Error bit in the SLAVE status register. SLAVEs shall execute all the commands defined below regardless of the state of the BUSY bit in the SLAVE Status Register. The results of the following commands shall be reflected in the returned ACKNOWLEDGE packet (if ACKNOWLEDGE has been requested) except for the Read Status command, see Section "5.3.3 Read Status Register" on page 27.

5.3.1 Reset SLAVE. This command shall cause all TM-Bus SLAVEs to come to an error-free quiescent state and all internal registers, counters and buffers to be brought to a known initial state such that each SLAVE is capable of receiving and executing commands. In response to a Reset SLAVE command, a SLAVE shall reset (release) all SLAVE Status Register bits, reset the SLAVE's multicast select group mode shall be reset to '00', and disable interrupts.

In response to a broadcast or multicast Reset command, a SLAVE shall set the Broadcast/Multicast Received bit.

5.3.2 Initialize Module. In response to an Initialize Module command, a SLAVE shall initialize the application side of the module bringing required registers to a pre-defined state.

5.3.3 Read Status Register. Upon the receipt of a NON-broadcast/multicast read status command, the SLAVE shall return the ACKNOWLEDGE packet, which includes the current eight (8) bit SLAVE Status Register contents, and then reset (release) the Bus Error, Event Occurrence , and Broadcast Received bits. Resetting the User-Defined bit in the SLAVE Status Register shall be optional.

If a broadcast/multicast read status command is received, the SLAVE shall not transmit any response over the SLAVE DATA line. It shall reset the Bus Error, Event Occurrence, and Broadcast Received bits. Resetting the User Defined bit in the SLAVE Status Register shall be optional.

5.3.4 CONTEND for Bus. This command shall cause SLAVEs to CONTEND for the bus as described in Section "5.2.9 TM-Bus Contention" on page 25.

5.3.5 Enable Interrupt. Upon receiving this command, a SLAVE shall set the Interrupt Enable bit in the SLAVE Status Register, thereby allowing the SLAVE to interrupt during IDLE or PAUSE states.

5.3.6 Disable Interrupt. Upon receiving this command, a SLAVE shall reset (release) the Interrupt Enable bit in the SLAVE Status Register, thereby preventing the SLAVE from interrupting.

5.3.7 Multicast Select 0. Upon receiving this command, a SLAVE shall enter multicast group 0 by resetting the SLAVE Status Register Multicast Select bits to '00'. This shall enable the SLAVE to respond to command headers with an address field equal to 'FC' (HEX).

5.3.8 Multicast Select 1. Upon receiving this command, a SLAVE shall enter multicast group 1 by setting the SLAVE Status Register Multicast Select bits to '01'. This shall enable the SLAVE to respond to command headers with an address field equal to 'FD' (HEX).

5.3.9 Multicast Select 2. Upon receiving this command, a SLAVE shall enter multicast group 2 by setting the SLAVE Status Register Multicast Select bits to '10'. This shall enable the SLAVE to respond to command headers with an address field equal to 'FE' (HEX).

5.3.10 Multicast Select 3. Upon receiving this command, a SLAVE shall enter multicast group 3 by setting the SLAVE Status Register Multicast Select bits to '11'. This shall enable the SLAVE to respond to command headers with an address field equal to 'FF' (HEX).

Command Field (MSB) (LSB) (8) (2)	Command
0000000	READ STATUS
0000001	INITIALIZE MODULE
0000010	RESET SLAVE
0000011	CONTEND FOR BUS
0000100	MULTICAST SELECT 0
0000101	MULTICAST SELECT 1
0000110	MULTICAST SELECT 2
0000111	MULTICAST SELECT 3
0001000	ENABLE INTERRUPT
0001001	DISABLE INTERRUPT
0001010	RESERVED
0001011	RESERVED
0001100	RESERVED
0001101	RESERVED
0001110	RESERVED
0001111	RESERVED

Figure 19. Standard Commands

5.4 TM-Bus Synchronization/Initialization. The bus shall be initialized when both the MASTER DATA and CONTROL lines are simultaneously released, forcing the bus into the IDLE state. All SLAVEs on the bus shall then be capable of transactions over the bus. If desired, the command 'Reset SLAVE' may then be broadcast to bring all bus lines and SLAVEs to an error-free quiescent state. When a SLAVE receives the Reset command, it resets all the SLAVE Status Register bits, resets its multicast select mode to '00', and disables SLAVE interrupts.

In response to a broadcast or multicast Reset command, a SLAVE shall set the Broadcast/Multicast Received bit.

5.5 TM-Bus Mastership. The TM-Bus shall have single MASTER operations. This specification shall not preclude the ability for systems to have more than one MASTER and a method to switch mastership of the bus independent of the four signal lines defined in this specification.

5.6 TM-Bus Error Handling. SLAVE(s) that detect a parity error in a HEADER packet shall ignore the command, set the Bus Error bit in the SLAVE status register and signal an interrupt as described in Section "5.2.8 TM-Bus Interrupts" on page 24. When a parity error is detected by the addressed SLAVE while receiving DATA packets, the SLAVE shall set the Bus Error bit and signal an interrupt.

Stuck-at-0 bus conditions are detected by the odd packet parity scheme as described in Section "5.2.2.2 Packet Parity" on page 15. Stuck-at-1 bus conditions shall be detected as an illegal command as described in Section "5.3 Command Definitions" on page 27.

To insure error-free reception during broadcast or multicast, the read status register command should be broadcast or multicast first to clear each SLAVE's broadcast/multicast received bit. After broadcast or multicast of data or a command, the broadcast/multicast received bit should be checked by reading the status register of each SLAVE one at a time.

5.7 TM-Bus Testing. On-line testing of the bus is performed as a result of its normal operation. Off-line or power-up testing may be accomplished through the use of bus exercise routines and bus wrap/hand-shaking tests. The MASTER shall be able to send bad parity or set any SLAVE's Event Occurrence bit and check for proper SLAVE response utilizing user definable commands for test flexibility.

6 NOTES

Any comments should be submitted to:

J.P. Letellier
Naval Research Lab
Code 5305
Washington, D.C. 20375
phone (202) 767-2937

APPENDIX I

10 GLOSSARY

EOM	-	End of Message
Hz	-	Hertz
LSB	-	Least Significant Bit
MHz	-	Megahertz, 1 million cycles per second
mA	-	Milliamperes, 1 thousandth of an Ampere
MID	-	Module Identification
MIP	-	Module Identification Parity
MSB	-	Most Significant Bit
nh	-	Nanohenry, 1 billionth of a Henry
ns	-	Nanosecond, 1 billionth of a Second
pF	-	Picofarad, 1 trillionth of a Farad
TBD	-	To Be Defined
Tf	-	Fall Time
Th	-	Hold Time
TM-Bus	-	Test and Maintenance Bus
Tpdhl	-	Propagation Delay, high-to-low
Tpdlh	-	Propagation Delay, low-to-high
Tr	-	Rise Time
Ts	-	Set-up Time
uA	-	micro Amperes, 1 millionth of an Ampere
Vih	-	High-level Input Voltage
Vil	-	Low-level Input Voltage
Vol	-	Low-level Output Voltage

VHSIC Phase 2 INTEROPERABILITY STANDARDS

Appendix D

ETM-Bus SPECIFICATION

November 9, 1987

Version 3.0

IBM

Honeywell

TRW

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

LIST OF ILLUSTRATIONS

1 SCOPE

1.1 Scope. This specification establishes the electrical, functional and performance requirements for the set of signal lines that constitute the Element Test and Maintenance Bus (ETM-Bus).

1.2 Purpose. The purpose of this standard is to establish requirements for the ETM-Bus and facilitate interoperability of VLSI chips which use the ETM-Bus.

1.3 Intended Application. The ETM-Bus is intended as a serial path for test and maintenance control and data information at the chip level.

2 APPLICABLE DOCUMENTS

2.1 Government Documents. The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- VHSIC Phase 2 INTEROPERABILITY STANDARDS TM-Bus SPECIFICATION, Version 3.0 dated November 9, 1987.
- VHSIC Phase 2 ELECTRICAL INTERFACE SPECIFICATION, Version 1.3 dated March 20, 1986.

2.2 Non-Government Documents. The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- None.

3 DEFINITIONS

The definitions listed here shall apply to the ETM-Bus and the VLSI chips which interface to it.

3.1 Item Definition. The ETM-Bus is a communications media which transfers bit serial data between a 'MASTER' device (CONTROLLER) and up to 32 logical 'SLAVE' elements interfacing to a single ETM-Bus CONTROLLER.

ETM-Bus elements implement the ETM-Bus protocol and meet all requirements of this specification.

Figure 1 on page 2, illustrates the ETM-Bus and ETM-Bus elements. Conceptually, each element consists of a device which performs the application specific function of the chip and a bus interface which implements the ETM-Bus master-slave communications protocol.

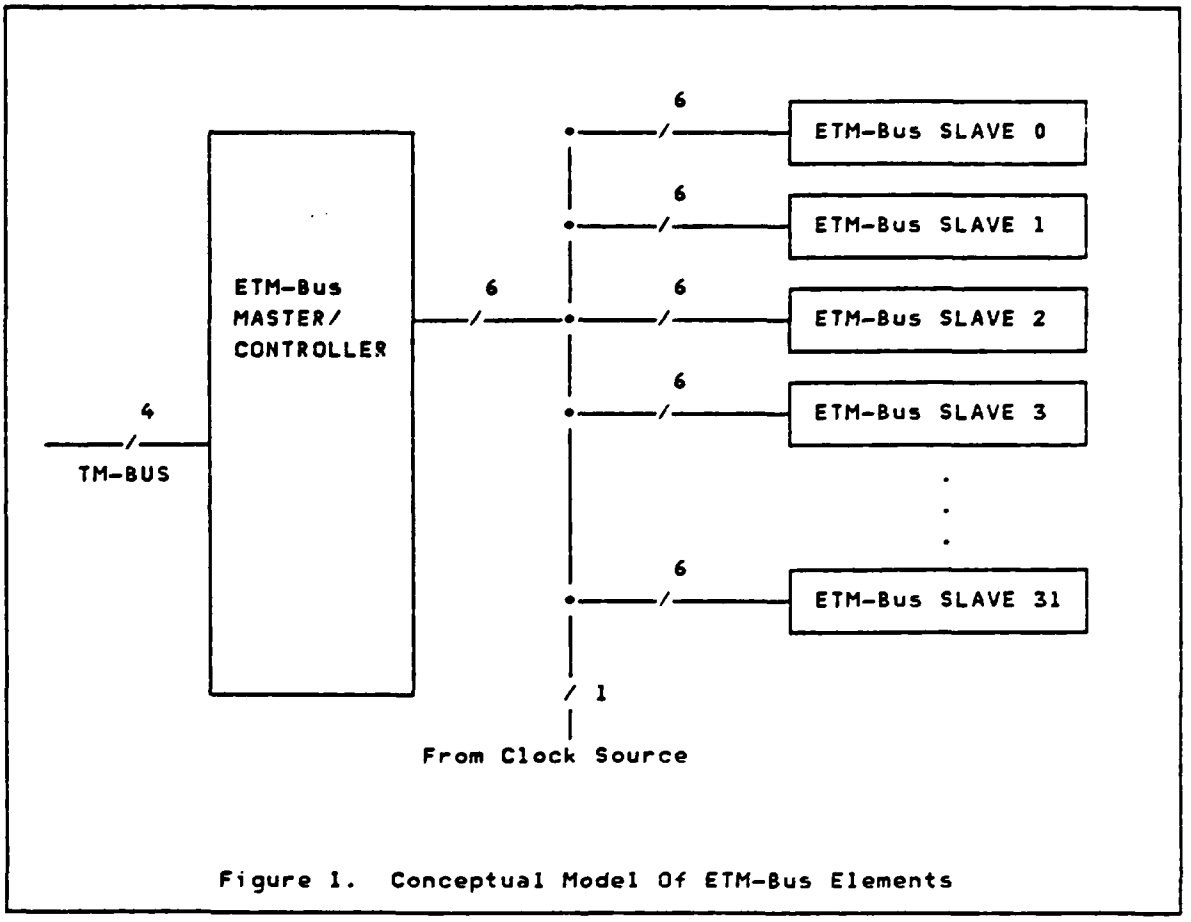


Figure 1. Conceptual Model Of ETM-Bus Elements

3.2 Term Definitions.

assert — The action of changing the state of a bus signal line from released, logic 0, to asserted, logic 1, or of ensuring that the line remains in the asserted state.

asserted	The logic 1 state of a bus signal line.
bus master	The device in control of the bus.
element	A single VLSI integrated circuit which interfaces to the ETM-Bus.
logic 0	The least positive of the two bus states for an active high bus signal line, or the more positive of the two bus states for an active low bus signal line.
logic 1	The more positive of the two bus states for an active high bus signal line, or the least positive of the two bus states for an active low bus signal line.
message	A set of sequences consisting of instructions or scan data.
release	The action of ceasing to assert a logic 1 on a bus signal line. The action of releasing a signal line produces a change in the state of the signal line only if no element is asserting that signal.
released	The logic 0 state of a bus signal line produced when no element asserts the signal associated with that line.
sequence	A transaction comprising a number of ordered transfers performing one intended function.
slave	An element which does not have control of the bus, and is selected by the master to participate in a sequence.
transfer	A set of elemental operations on the bus which result in the communication of bit serial datum units between the current bus master and the selected slave(s). A serial datum unit is 1 bit. See sequence.
wired-ORed	A hardwired OR function.

4 PHYSICAL LAYER

4.1 Introduction. The physical layer of the ETM-Bus is specified herein. The bus signal lines required to implement the bus, the electrical characteristics of the bus interfaces,the allowable bus configurations, and timing definitions are presented.

4.2 Line Definition. The ETM-Bus signal and clock lines are defined in this section.

4.2.1 Nomenclature. Lines shall be designated by name. When a set of related bits are represented by the same name, the bits within the set shall be differentiated by number with the most significant (MSB) bit numbered 0. All fields shall be referred to by their bit position within a data word transferred over the ETM-Bus. All ETM-Bus signals are defined in the following paragraphs. A "-" symbol associated with a signal means that the signal is active low.

4.2.2 ETM-Bus Signal Definition. There shall be a minimum of six (6) signal types that make up the ETM-Bus as shown in Figure 2 on page 5. Additional lines, to accommodate enhanced application requirements, are not precluded. SELECT and INTERRUPT are negative logic and all other signals shall use positive logic.

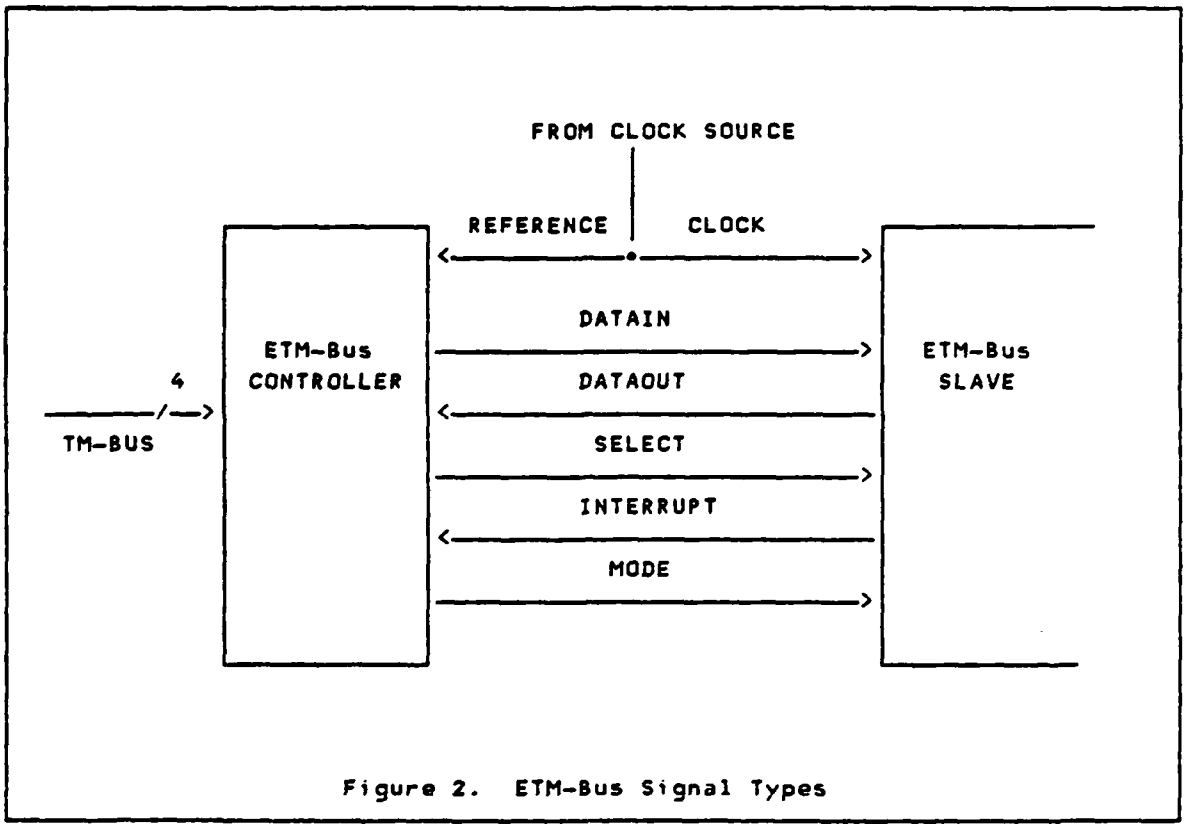


Figure 2. ETM-Bus Signal Types

4.2.2.1 ETM-Bus CLOCK Signal Definition. All data transfer operations shall be synchronous with the REFERENCE CLOCK (REF CLK) signal. All bus activity shall be relative to the high-to-low transition of the REFERENCE CLOCK. All data transfers shall occur as shown in Figure 4 on page 7. The CLOCK signal shall be single phase. The ETM-Bus interface should support the full range of clock frequencies from zero (0) to 6.25 MHz.

4.2.2.2 ETM-Bus DATAIN Signal Definition. The ETM-Bus DATAIN signal shall be a single unidirectional line into the SLAVE. Instruction data and scan data shall be transmitted to the SLAVEs over the DATAIN line. In the ring configuration, shown in Figure 5 on page 8, the DATAIN signal shall be sourced from either the CONTROLLER or another SLAVE. In the star configuration, shown in Figure 6 on page 9, the DATAIN signal shall be sourced from the CONTROLLER.

4.2.2.3 ETM-Bus DATAOUT Signal Definition. The ETM-Bus DATAOUT signal shall be a single unidirectional line from a SLAVE. In the star configuration, the DATAOUT signal shall be transmitted from the SLAVE to the CONTROLLER. In the ring configuration, the DATAOUT signal shall be transmitted from the SLAVE to either the CONTROLLER or the DATAIN pin of another SLAVE.

The DATAOUT line shall support three-state operation. The DATAOUT signal shall be in the high-impedance state when inactive (e.g., not in a logic 1 or 0 state).

4.2.2.4 ETM-Bus SELECT Line Signal Definition. The SELECT signal line (-SEL) shall be unidirectional from the CONTROLLER to the SLAVEs. The SELECT signal line defines when data transfer operations shall occur. SELECT shall be asserted (low) one cycle before instruction or scan data is serially transferred across the DATA lines. SELECT shall be released one cycle before the end of a data transfer.

In a ring bus structure, where all SLAVEs share a common SELECT, all SLAVEs are selected simultaneously, (see Figure 5 on page 8). In a star bus structure, all SLAVEs shall have a separate SELECT line (see Figure 6 on page 9).

4.2.2.5 ETM-Bus INTERRUPT Signal Definition. INTERRUPT (-INT) shall be unidirectional from the SLAVEs to the CONTROLLER. The INTERRUPT line shall be asserted (low) to indicate that an event (such as an error or other predetermined condition) has occurred. The INTERRUPT line shall remain asserted until the interrupt is serviced. The INTERRUPT line shall support a wired-OR configuration. The INTERRUPT line may operate asynchronously.

4.2.2.6 ETM-Bus MODE Signal Definition. The ETM-Bus MODE signal shall be unidirectional from the CONTROLLER to the SLAVE. MODE shall be used to establish the type of operation that is performed when SELECT is being asserted (see Figure 3 on page 6).

MODE	OPERATION
0	INSTRUCTION/STATUS
1	SCAN

Figure 3. ETM-Bus MODE Line Definition

MODE shall be valid one cycle before instruction or scan data is serially transferred across the DATA lines and remain stable for the length of the transfer. The MODE line shall stay valid as long as SELECT is asserted (see Figure 4 on page 7).

The use of the ETM-Bus shall be defined by the level of the SELECT line and the MODE line. Two types of data transfer operations can take place on the ETM-Bus as specified in Figure 3 on page 6. Data transfers shall end by no

more than one (1) clock cycle after the SELECT line is released (high).

4.3 ETM-Bus Timing Relationships. All ETM-Bus signals shall be referenced to the falling edge of the REFERENCE clock (see Section "4.5 ELECTRICAL REQUIREMENTS" on page 10 for details). Figure 4 on page 7 depicts the timing relationship of the ETM-Bus signals.

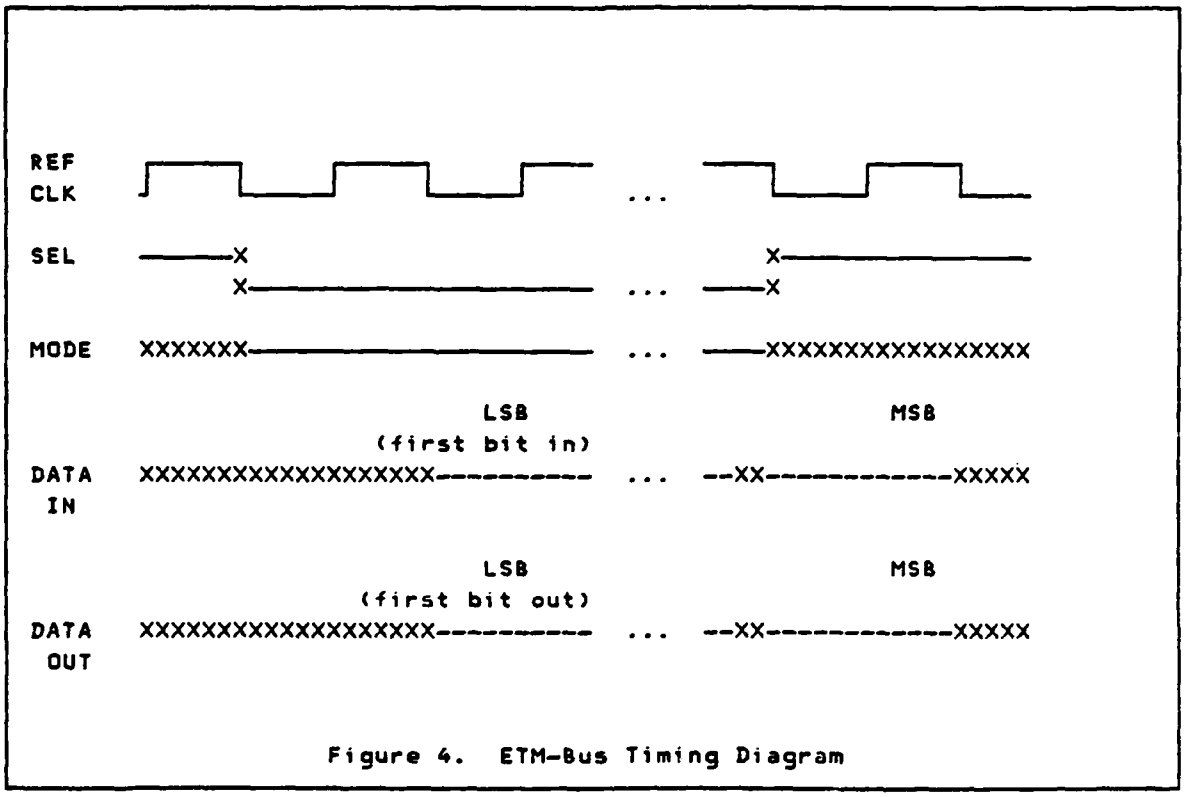


Figure 4. ETM-Bus Timing Diagram

4.4 ETM-Bus Configurations. The allowable ETM-Bus signal configurations are specified in this section.

4.4.1 ETM-Bus Ring Bus Configuration. The ETM-Bus shall be capable of supporting the ring bus structure shown in Figure 5 on page 8. The SELECT line shall be connected to all SLAVEs within the loop and all chips are enabled together for instruction/status and data/scan operations.

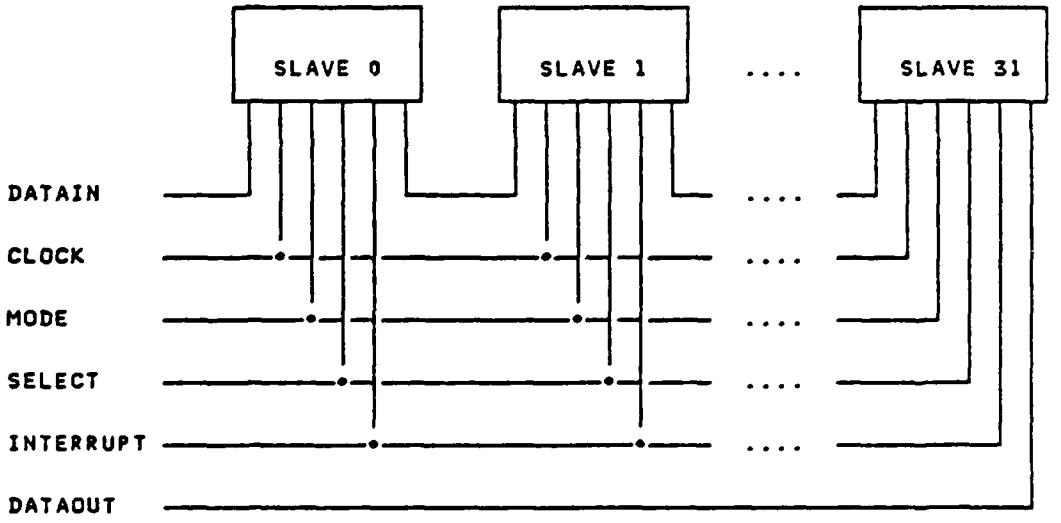


Figure 5. ETM-Bus Ring Bus Structure

4.4.2 ETM-Bus Star Bus Configuration. The ETM-Bus shall be capable of supporting the star bus structure shown in Figure 6 on page 9 which uses multiple SELECT lines. The SELECT lines may be used to select single SLAVEs independently for instruction/status and data/scan operations, or multiple SLAVEs may be selected for instruction/status operations. The use of multiple INTERRUPT and/or DATAOUT lines in the star bus structure are not precluded.

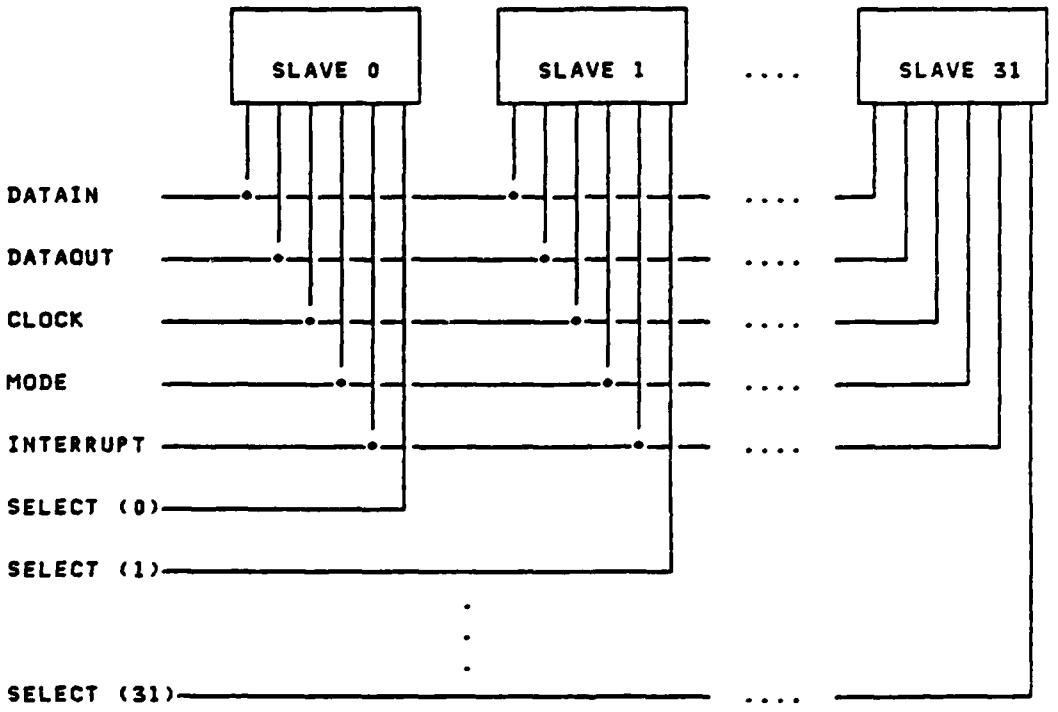


Figure 6. ETM-Bus Star Bus Structure

4.5 ELECTRICAL REQUIREMENTS

Electrical requirements defining the device output signals, the device input signals, and the ETM-Bus clock shall be specified herein.

4.5.1 DC Requirements for Device and Bus Interface

DC requirements for device and bus interface shall be as specified in Version 1.3 of the VHSIC Phase 2 Electrical Interface Specification, dated March 20, 1986. Refer to Section "4.2 DC Electrical Limits for Inputs and Outputs" on page 4-1.

4.5.2 AC Requirements for Device and Bus Interface

4.5.2.1 Reference Voltage. The reference signal voltage (Vref) for timing shall be +1.5 volts. The reference clock voltage for timing shall be +1.5 volts.

4.5.2.2 Clock and Signal Inputs to Devices

4.5.2.2.1 Set-up Time. The minimum time that each input signal shall be uniquely above or below the reference signal voltage prior to the high-to-low transition of the REFERENCE CLOCK (set-up time, Ts) is 15 ns. See Figure 7 on page 11.

4.5.2.2.2 Hold-Time. The minimum time that each input signal shall be uniquely above or below the reference signal voltage following the high-to-low transition of the REFERENCE CLOCK (hold time, Th) is 15 ns. See Figure 7 on page 11.

4.5.2.2.3 Configuration Dependent Parameters

4.5.2.2.3.1 Propagation Delay. The propagation delay time (Tp) shall be measured from the high-to-low transition of the REFERENCE CLOCK voltage to the reference signal voltage of the designated element output signal as shown in Figure 8 on page 11.

The minimum propagation delay for the element output signal, DATAOUT, shall be specified.

The maximum propagation delay for the element output signal, DATAOUT, shall be specified.

4.5.2.2.3.2 Bus Settling Time. The bus settling time (Tbs) shall be specified, see Figure 9 on page 12.

4.5.2.2.3.3 Clock Skew. The maximum clock skew (Tcs) between elements on the ETM-Bus shall be specified, see Figure 9 on page 12.

4.5.2.2.3.4 Reference Clock. The reference clock cycle time (Tclk) shall be specified, see Figure 9 on page 12, (e.g., 6.25 Mhz/160 ns).

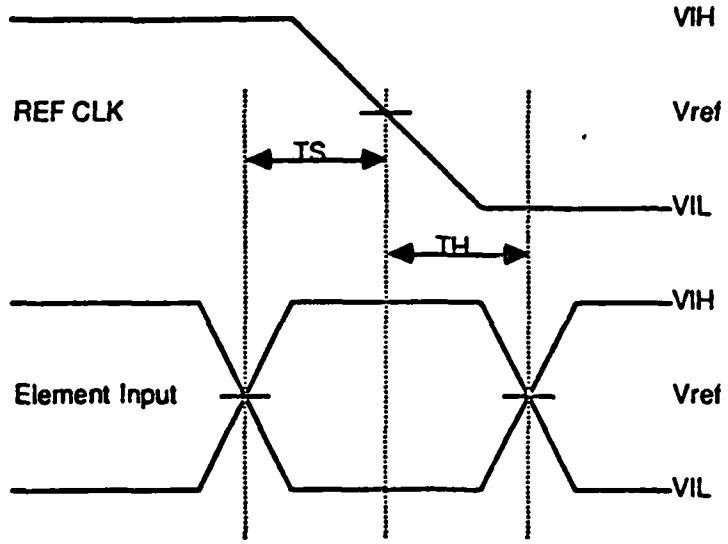


Figure 7. Set-Up and Hold Time

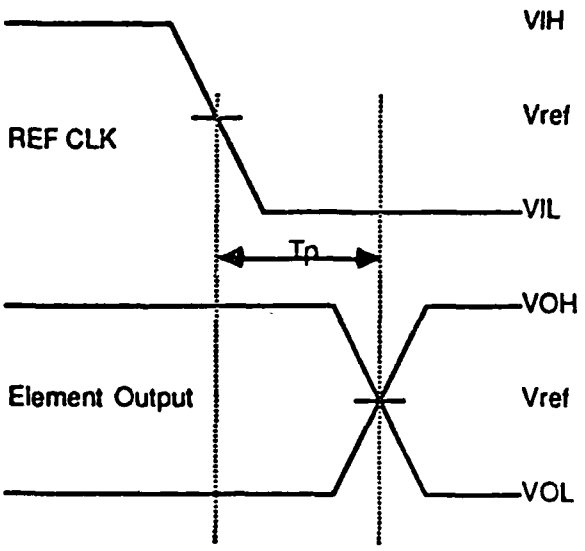


Figure &. Propagation Delay

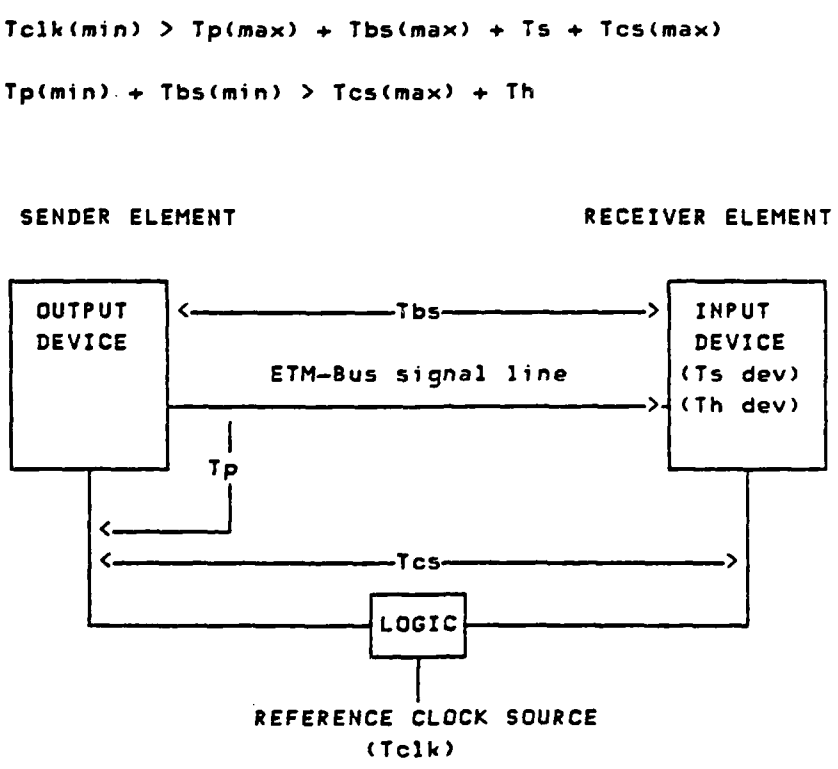


Figure 9. ETM-Bus Interface Parameters

4.5.3 Bus Clock Requirements

4.5.3.1 Rise and Fall Time. The rise time (Tr) and fall time (Tf) of the REFERENCE CLOCK shall be the transition time from the voltage change of VIH to VOL. Both Tr and Tf shall be less than or equal to 4 ns.

4.5.3.2 Duty Cycle. For single clock implementations, the ratio of the REFERENCE CLOCK high state duration to the REFERENCE CLOCK period measured at the reference clock voltage shall not be less than 0.45 nor greater than 0.55.

5 DATA LINK LAYER

5.1 Introduction. The Element Test and Maintenance Bus (ETM-Bus) shall be a channel for control and data information flow between a module maintenance controller and the individual elements (e.g. VHSIC Phase 2 chips) within the module. The module maintenance controller may be implemented as a Test and Maintenance Bus (TM-Bus) SLAVE as described in the TM-Bus specification. The module maintenance controller shall be referred to as the CONTROLLER and all other elements on the ETM-Bus shall be referred to as SLAVEs. The information transferred and the scheduling of data and instructions is system dependent and is not addressed in this specification. Figure 10 on page 13 summarizes the ETM-Bus design parameters and characteristics.

o Performance Characteristics	o Protocol Characteristics
- 6.25 MHz clock (Typical)	- Each ETM-Bus logically supports up to 32 SLAVE devices
- Unidirectional data lines	
- Minimum of 6 pin bus signals	- Supports ring or star configurations
- TTL compatible	- Interrupt Capability

Figure 10. ETM-Bus Design Parameters and Characteristics

This section details the protocol requirements for all elements attached to the ETM-Bus.

5.2 Operation. The element interface to the ETM-Bus shall appear to the CONTROLLER as a shift register when the CONTROLLER selects the element for operation. All data shall be transmitted to and from the CONTROLLER LSB first.

Figure 11 on page 14 shows a conceptual diagram of the element interface to the ETM-Bus. When the CONTROLLER selects the element for instruction/status operation, the element configures the instruction/status register as a shift register, shifting in a new instruction from the CONTROLLER as it shifts out the contents of a selected register (Status Register, Register A, Register B, etc.) to the CONTROLLER. When the CONTROLLER selects the element for data scan operations, the element shifts out the contents of either a selected scan path (see Section "5.2.4.2 Addressing of Scan Paths" on page 17) or the Bypass Bit (see Section "5.2.4.3 Bypass Bit with Ring Bus Implementation" on page 17). When the element is deselected, the ETM-Bus interface controls the element testability features by executing the instruction previously-received from the CONTROLLER. The interface control and decode

logic is used to decode instructions and to store control information that must remain stable as new instructions are shifted into the interface. The control information is used to select the parallel input to the instruction/status register, to determine the output for the ETM-Bus signal DATAOUT, and to control element testability features.

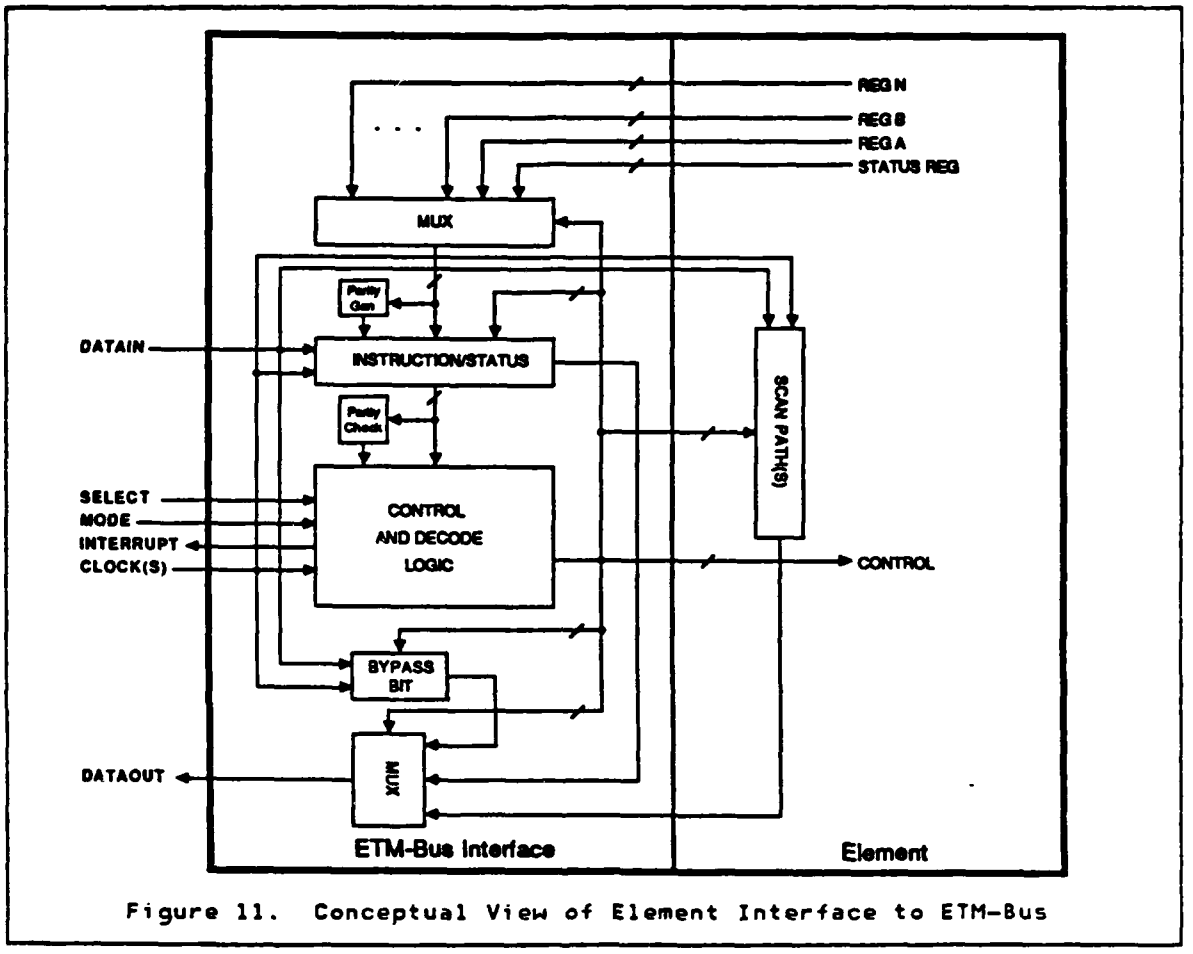


Figure 11. Conceptual View of Element Interface to ETM-Bus

5.2.1 ETM-Bus States. All valid operations on the ETM-Bus shall be either instruction/status or scan operations as determined by the MODE signal line (see Section "4.2.2.6 ETM-Bus MODE Signal Definition" on page 6). Figure 12 on page 15 depicts the protocol and state transition sequences of the ETM-Bus. It is required that there be a minimum of one IDLE state between SCAN operations and a minimum of two IDLE states between instruction operations. Interrupts from SLAVEs are allowed over the INTERRUPT signal line during any bus state.

For every bit transferred into a SLAVE on DATAIN, a bit shall be transferred out of a SLAVE on DATAOUT. The CONTROLLER shall begin a transfer by asserting the SELECT line, with the MODE line valid, on the cycle prior to a data

transfer. The CONTROLLER shall signal the end of an operation by releasing SELECT on the cycle prior to the end of the data transfer. SELECT shall stay released at least two cycles between SCAN operations and at least three cycles between instruction operations.

The PRE-SCAN (S1) and PRE-INSTRUCTION (S2) states allow the SLAVEs to receive an indication of a bus state change before actual transmission of data begins. The POST-SCAN (S5) and POST-INSTRUCTION (S6) states correspond to the transfer of the last bit of data as the SLAVE receives notification of the change in the bus state.

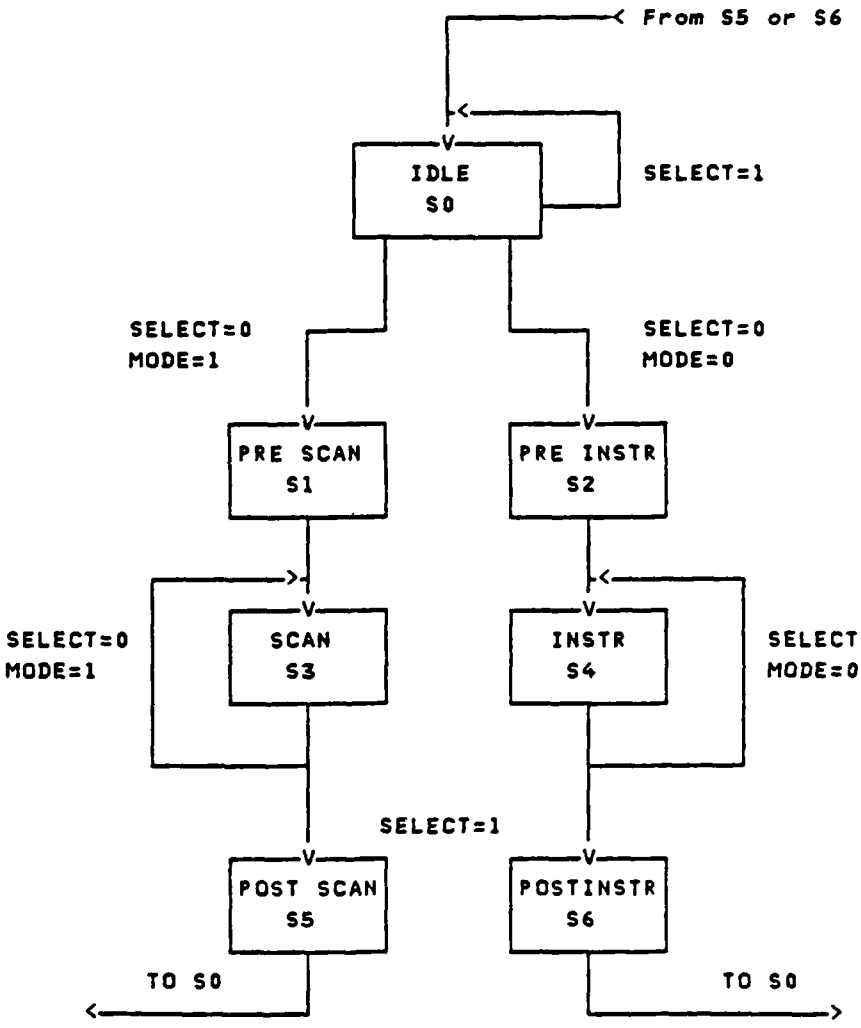


Figure 12. ETM-Bus Protocol State Diagram

5.2.2 SLAVE Device Identification. In a ring structure, the SLAVE ID shall be determined by the element's position in the ring. In a star structure, the SLAVE ID shall be determined by the connections of the CONTROLLER SELECT lines.

The CONTROLLER shall ensure that SLAVEs receive valid data during a transfer operation.

5.2.3 ETM-Bus Instruction/Status Protocol

5.2.3.1 Introduction. An instruction/status protocol sequence shall be used to send an instruction to a SLAVE and to receive SLAVE status or scan data. When the MODE Signal line is a logic 0, the selected SLAVE(s) shall appear as a serial shift register. When SELECT is asserted and MODE is a logic 0, the contents of the virtual instruction/status register are shifted. Data is input to the register from the SLAVE's DATAIN signal line and output from the register to the ETM-Bus DATAOUT signal line. All SLAVEs shall have the ability to perform this shift operation during functional (on-line) chip operation without affecting chip operation.

The contents of the virtual instruction/status register may contain the current element status information at the time of the shift operation. If the interface supports this read status operation, the status information can be shifted around the loop and read by the CONTROLLER.

5.2.3.2 Instruction Execution. When the SLAVE(s) is deselected (when SELECT is released), the SLAVE shall execute the current instruction in the instruction/status register within three (3) clock cycles.

A parity check shall be performed on the instruction before execution. If an error is detected, the instruction shall not be executed and an interrupt shall be sent to the CONTROLLER as described in Section "5.2.7 Interrupts" on page 17.

The instruction/status protocol shall not interfere with the normal function of the SLAVE application.

5.2.3.3 Ring Bus Implementation. When in the ring bus configuration, all SLAVEs within the ring are selected simultaneously for an instruction/status operation. A sequence of instruction(s) are shifted in the DATAIN line of the first SLAVE in the ring and the status word(s) are shifted through the ring and out of the DATAOUT line of the last SLAVE in the ring. Instruction execution shall be performed as described in Section "5.2.3.2 Instruction Execution" on page 16.

5.2.4 ETM-Bus Scan Protocol

5.2.4.1 Introduction. To send data from a SLAVE or to receive data from a SLAVE, a scan data protocol sequence shall be used. In the scan mode, the selected SLAVE(s) shall appear to the bus as a serial shift register (referred to herein as scan path) with the DATAIN line as the serial input and the DATAOUT line as the serial output of the scan path. Therefore, when SELECT is asserted and MODE is a logic 1, the contents of the virtual scan path register shown in Figure 11 on page 14 are shifted. SLAVEs may require that the chip be taken off-line to perform this operation as it may affect the SLAVE application operation.

5.2.4.2 Addressing of Scan Paths. The virtual scan path shift register may be implemented as a number of parallel scan paths of varying lengths. The selection of any serial scan path contained with the SLAVE shall be performed by instructions previously received by the SLAVE. Use of SLAVE application scan paths may necessitate disabling SLAVE functions (off-line operation) during certain scan sequences.

5.2.4.3 Bypass Bit with Ring Bus Implementation. All SLAVEs shall support a one bit scan path which shall be known as the "Bypass" configuration bit. This path shall be selected in the same way as any other application scan path. This "bypass" scan path will enable shorter access to individual chips which require repeated scans when a ring bus structure is used by allowing selected chips to be "bypassed" during scan operations. All SLAVEs shall have the ability to scan the "bypass" scan path during functional (on-line) chip operation without affecting chip operation.

5.2.5 Instruction Register. The instruction register is loaded serially as described in Section "5.2.3 ETM-Bus Instruction/Status Protocol" on page 16. The instruction register is 17 bits in length which includes 16 bits of data and 1 bit odd parity.

5.2.6 Status Register. The status register is read serially as described in Section "5.2.3 ETM-Bus Instruction/Status Protocol" on page 16. The status register is 17 bits in length which includes 16 bits of status and 1 bit odd parity.

5.2.7 Interrupts. Interrupts are indicated by a SLAVE asserting the INTERRUPT line until the CONTROLLER services the interrupt (see Section "5.5 ETM-Bus Initialization" on page 18).

5.3 SLAVE Operation Sequences. Commands are dependent on element implementation within a system and are not specified herein. A recommended list of SLAVE operation sequences follows:

- NOOP- executes no operation.
- ECHO- data sent in instruction word is 'echoed' back on following instruction.
- WRITE REGISTER - write data to specified SLAVE register.
- READ REGISTER - read specified register into SLAVE instruction/status register.
- RESET - places element logic into a predefined state.
- READ STATUS - loads status bits into SLAVE instruction/status register.
- ENABLE/DISABLE CHIP OUTPUTS
- ENABLE/DISABLE CHIP CLOCKS
- ENABLE/DISABLE CHIP INTERRUPTS

5.4 ETM-Bus Control. The ETM-Bus shall have single CONTROLLER operations. This specification shall not preclude the ability for systems to have more than one CONTROLLER and a method to switch mastership of the bus independent of the signal lines defined in this specification.

5.5 ETM-Bus Initialization. It shall be the responsibility of the CONTROLLER to initialize the ETM-Bus. This shall be accomplished by releasing the SELECT line and holding the MODE line low (instruction mode).

5.6 ETM-Bus Error Handling and Recovery Definition. All instructions transferred from the CONTROLLER to a SLAVE shall have one odd parity bit in the most significant bit of each instruction shifted during instruction/status operations (i.e., 16 data bits plus 1 parity bit). The SLAVE shall perform parity checking at the completion of an instruction transfer. If a parity error is detected, the SLAVE shall not execute the instruction and shall assert an "Instruction Parity Error" bit in the SLAVE Status register and transmit an interrupt to the CONTROLLER.

The SLAVE Status Register shall have one odd parity bit in the most significant bit of each status word (i.e., 16 data bits plus 1 parity bit) which is transferred to the CONTROLLER with SLAVE Status during Status reads.

5.7 ETM-Bus Testing. On-line testing of the bus is performed as a result of its normal operation. Off-line or power-up testing may be accomplished through the use of bus exercise routines and bus wrap/hand-shaking tests. The CONTROLLER shall be able to send bad parity to check the SLAVE's inter-

rupt logic and check for proper SLAVE response utilizing user definable commands for test flexibility.

6 NOTES

Any comments should be submitted to:

J.P. Letellier
Naval Research Lab
Code 5305
Washington, D.C. 20375
(202) 767-2937

APPENDIX I

10 GLOSSARY

EOM	-	End of Message
ETM-Bus	-	Element Test and Maintenance Bus
LSB	-	Least Significant Bit
MHz	-	Megahertz, 1 million cycles per second
mA	-	Milliamperes, 1 thousandth of an Ampere
MSB	-	Most Significant Bit
nh	-	Nanohenry, 1 billionth of a Henry
ns	-	Nanosecond, 1 billionth of a second
pf	-	Picofarad, 1 trillionth of a Farad
TBD	-	To Be Defined
Tbs	-	Bus Settling Time
Tclk	-	Reference Clock Cycle Time
Tcs	-	Clock Skew
Tf	-	Fall Time
Th	-	Hold Time
Tp	-	Propagation Delay
Tr	-	Rise Time
Ts	-	Set-up Time
TTL	-	Transistor-Transistor Logic
uA	-	micro Amperes, 1 millionth of an Ampere
Vref	-	Reference Voltage

VHSIC Phase 2 INTEROPERABILITY STANDARDS

Appendix E

ELECTRICAL INTERFACE SPECIFICATION

January 21, 1988

Version 2.4

IBM

Honeywell

TRW

CONTENTS

ABSTRACT

This specification defines the electrical interface standard for VHSIC Phase 2 integrated circuits. The use of a standard electrical interface will simplify the design of military systems using VHSIC Phase 2 integrated circuits from multiple sources.

PREFACE

This document was prepared by IBM, Honeywell and TRW in partial fulfillment of Contract Data Requirements List (CDRL) item A011 for work being performed under VHSIC Phase 2 Submicrometer Technology Development contracts DAAK20-85-C-0367, F33615-84-C-1500 and N00039-85-C-0111, respectively.

Section 1

SCOPE

1.1 SCOPE.

This specification states the electrical interface and clock requirements for VHSIC Phase 2 integrated circuits.

1.2 PURPOSE.

The purpose of this specification is to establish an electrical interface and clock standard that facilitates interoperability of VHSIC Phase 2 integrated circuits.

1.3 INTENDED APPLICATION.

This specification applies to the following VHSIC Phase 2 electrical interfaces:

- Power supplies.
- DFNCLK and SYSCLK clock signals at the chip inputs. Any other SYSCLK synchronous clocks required by a VHSIC chip or chip set will be generated by that chip or chip set from the DFNCLK and SYSCLK. The manufacturer of the chip or chip set may elect to provide additional JAN qualified clock generating chips in the set to provide additional clocks if the design requires them, but they must also require only the DFNCLK and SYSCLK clocking inputs.
- Electrical logic levels for point-to-point binary data interchange.

Section 2

APPLICABLE DOCUMENTS

2.1 GOVERNMENT DOCUMENTS.

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- VHSIC Phase 2 Interoperability Standards ETM-Bus Specification.

2.2 NON-GOVERNMENT DOCUMENTS.

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- None

Section 3

DEFINITIONS

The definitions listed herein shall apply.

absolute maximum rating — Phrase used to identify a stress condition beyond which the integrated circuit may be permanently damaged. Functional operation of the integrated circuit under this stress condition is not implied. Long term reliability may be reduced, even if damage is not immediately observable.

operating conditions — Phrase used to specify the range of a parameter over which functional operation of the integrated circuit is provided. Exceeding the operating conditions for more than one minute may affect the reliability of the integrated circuit.

signal midpoint — Signal midpoint is defined to be that voltage which is the average of the maximum input low voltage and minimum input high voltage as defined in sections 4.2.2.2 and 4.2.2.3.

clock skew — Clock skew is defined to be the delay between the falling edges of two synchronous signals as measured at the signal midpoint value as shown in Figure 1.

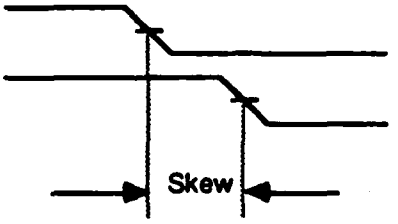


Figure 1: Clock Skew

duty cycle — The duty cycle shall be defined as the time the clock is higher than the midpoint value divided by the total clock period.

synchronous

Two signals are synchronous if one signal's frequency is an integer multiple of the other's frequency. There also exists a well defined phase relationship between the two signals, as shown in Figure 2, where the value t_d is constant.

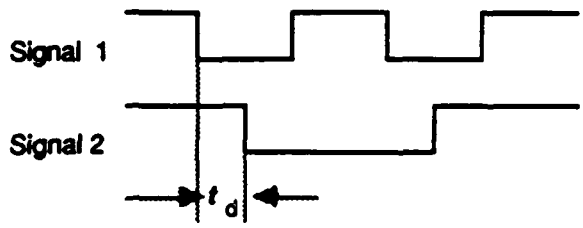


Figure 2: Timing Relationship of Two Synchronous Signals

Section 4

ELECTRICAL INTERFACE REQUIREMENTS

Electrical interface characteristics for VHSIC Phase 2 integrated circuits shall be as specified herein. All voltages shall be referenced to the integrated circuit ground. Current shall be conventional current with current into the integrated circuit expressed as a positive value.

4.1 POWER SUPPLY REQUIREMENTS

4.1.1 ABSOLUTE MAXIMUM RATING

The absolute maximum DC voltage applied to any +3.3 volt power input shall be greater than -0.3 volts and less than +5.0 volts.

The absolute maximum DC voltage applied to any +5.0 volt power input shall be greater than -0.5 volts and less than +7.0 volts.

4.1.2 OPERATING CONDITION

The DC voltage applied to any +3.3 volt nominal power input during operation shall be +3.3 volts ±5 percent (3.135 volts to 3.465 volts).

The DC voltage applied to any +5.0 volt nominal power input during operation shall be +5.0 volts ±10 percent (4.5 volts to 5.5 volts).

4.2 DC ELECTRICAL LIMITS FOR INPUTS AND OUTPUTS

4.2.1 ABSOLUTE MAXIMUM RATINGS

Unless otherwise specified, for those inputs and outputs using the +3.3 volt nominal supply, the absolute maximum DC voltage applied to any input or output shall be greater than -0.3 volts. The absolute maximum DC voltage shall be less than +4.2 volts. When the voltage is between 3.465 volts and 4.2 volts the absolute value of the current supplied shall be less than 4mA.

Unless otherwise specified, for those inputs and outputs using the +5.0 volt nominal supply, the absolute maximum DC voltage applied to any input or output shall be greater than -0.5 volts and less than +7.0 volts.

Note: Transient voltages may be more negative as specified in 4.2.2.3.

4.2.2 OPERATING CONDITIONS

4.2.2.1 Input And Output (I/O) Capacitance

Capacitance to ground and other lines shall be specified for each integrated circuit input and output.

4.2.2.2 High-level Input Voltage (VIH)

An input voltage of +2.0 volts or more shall be interpreted as a high level. For those inputs using the +3.3 volt nominal supply, unless otherwise specified, the DC input voltage shall not exceed the value of the +3.3 volt nominal supply. For those inputs using the +5.0 volt nominal supply, unless otherwise specified, the DC input voltage shall not exceed the value of the +5.0 volt nominal supply.

4.2.2.3 Low-level Input Voltage (VIL)

An input voltage of +0.8 volts or less shall be interpreted as a low level. The minimum DC input voltage shall be -0.3 volts. A minimum input voltage of -0.6 volts shall be permitted as a transient while ringing is clamped as long as the signal is within dc limits for the specified setup/hold time from section 4.3.2.6.

4.2.2.4 Leakage Current (IL)

Over the input voltage range of 0 volts to the specified maximum VIH, the input current into any input or output that is not driving the line shall be between -250 microamps and +20 microamps.

4.2.2.5 High-level Output Voltage (VOH)

For active pull-up outputs, the high-level output voltage shall be greater than +2.4 volts at a current of -0.4 milliamps.

Open collector and open drain outputs are permitted.

4.2.2.6 Low-level Output Voltage (VOL)

The low-level output voltage shall be less than +0.5 volts at current of +4.0 milliamps.

4.2.2.7 Low-level Sink Current (IOL)

The low-level output current drive capability shall be greater than or equal to +4.0 milliamps at an output voltage of +0.5 volts.

4.2.2.8 High-level Source Current (IOH)

The high-level output current drive capability shall be greater than or equal to -0.4 milliamps at an output voltage of +2.4 volts.

4.3 CLOCK REQUIREMENTS

4.3.1 GENERAL DESCRIPTION

Two clocks shall be distributed to the subsystems. They shall be referred to as SYSCLK and DFNCLK. These signals shall be synchronous to each other. It is intended that the internal chip clock(s) be derived from the signal SYSCLK.

The signal DFNCLK is intended to be used as a synchronization signal to keep the various chips and applications in step with one another. The DFNCLK signal can be used for REFCLK in the ETM-bus.

4.3.2 CHARACTERISTICS OF SYSCLK

4.3.2.1 Frequency

Submicron VHSIC chips shall be designed to operate with SYSCLK having a maximum frequency of at least 100MHz and a minimum frequency of 0 Hz. Systems utilizing VHSIC chips may be designed to utilize a SYSCLK anywhere in the range of 0 to 100MHz, provided the other provisions of this specification are met.

4.3.2.2 Duty Cycle

The duty cycle of SYSCLK shall be no less than 45% and no more than 55%, measured at the input pin.

4.3.2.3 Clock Rise Time

The maximum rise time of SYSCLK shall be the lesser of 15% of SYSCLK's period or 5nS, measured at the input pin. The rise time shall be measured from the 10% point of the voltage waveform to the 90% point.

4.3.2.4 Clock Fall Time

The maximum fall time shall be the lesser of 15% of SYSCLK's period or 5nS, at the input pin. The fall time shall be measured from the 90% point of the voltage waveform to the 10% point.

4.3.2.5 Trailing Edge Operation

All chip application registers shall be referenced to a falling edge of SYSCLK. Clock control logic may use a rising edge of SYSCLK where required. The appropriate edges from the set 0:15 shall be specified for each register. For example one register could be specified to trigger on edges 0,2,4...14. another might use edges 1,3,5 ... 15. See Figure 3.

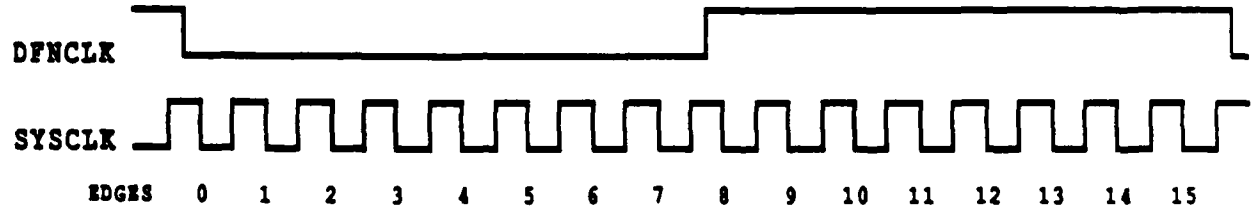


Figure 3: Clock timing

4.3.2.6 Setup Time/Hold Time Specification

All logic signal setup and hold times shall be specified with respect to the voltage midpoint on the falling edge of SYSCLK. Actual numbers are application dependent and will vary.

4.3.2.7 Clock to Output Specification

All logic signal clock to output delays shall be specified with respect to the voltage midpoint on the falling edge of SYSCLK. Actual numbers are application dependent and will vary.

4.3.2.8 DC Electrical Limits for SYSCLK

SYSCLK shall comply with Section 4.2.2.

4.3.3 Characteristics of DFNCLK

4.3.3.1 Frequency

The frequency of DFNCLK shall be the frequency of SYSCLK divided by 16.

4.3.3.2 Duty Cycle

The duty cycle of DFNCLK shall be no less than 49% and no more than 51%, measured at the input pin.

4.3.3.3 Clock Rise Time

The maximum rise time of DFNCLK shall be the lesser of 15% of SYSCLK's period or 5nS, measured at the input pin. The rise time shall be measured from the 10% point of the voltage waveform to the 90% point.

4.3.3.4 Clock Fall Time

The maximum fall time shall be the lesser of 15% of SYSCLK's period or 5nS, measured at the input pin. The fall time shall be measured from the 90% point of the voltage waveform to the 10% point.

4.3.3.5 Trailing Edge Operation

Registers clocked by DFNCLK shall be triggered on a falling edge of DFNCLK.

4.3.3.6 DC Electrical Limits for DFNCLK

DFNCLK shall comply with Section 4.2.2

4.3.4 Clock Skew Requirements

4.3.4.1 SYSCLK

The maximum clock skew for the SYSCLK signals between different chips in the subsystem, measured at the input pins should be the lesser of 5% of SYSCLK's period or 2nS, measured at the input pins.

4.3.4.2 SYSCLK to DFNCLK

The clocks should be distributed, such that measured at the chip's input pins, the falling edge of SYSCLK should occur after the falling edge of DFNCLK. The clock skew between DFNCLK and SYSCLK should be between 1 and 4 nS.

4.3.5 Relationship to REFCLK

4.3.5.1 Separate DFNCLK and REFCLK

Chips with separate DFNCLK and REFCLK inputs shall be capable of operating with both pins connected to DFNCLK.

4.3.5.2 DFNCLK as REFCLK

Chip specifications may require use of DFNCLK as the REFCLK signal in the ETM-bus. Furthermore if any chip on a particular ETM-bus uses DFNCLK as the REFCLK signal then all the chips on that bus must do so.

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

ARMY

U.S. Army Missile Command (1)
ATTN: Mr. Robert Hicks
AMSMI-RD-AS-RA
Redstone Arsenal, AL 35898

US Army ARDEC (1)
ATTN: Maj.erndon
SMCAR-FSF-BV
Dover, NJ 07801

US Army ARDEC (1)
ATTN: Mr. Joseph Lehman
SMCAR-FSF
Dover, NJ 07801

US Army LABCOM (1)
Army Research Office
ATTN: Dr. Jim Gault
PO Box 12211
Research Triangle Park, NC 27709-2211

US Army LABCOM (1)
ET&D Laboratory
ATTN: Mr. Harold Borkan
SLCET-DD
Fort Monmouth, NJ 07703-5000

US Army LABCOM (1)
ET&D Laboratory
ATTN: Dr. C. G. Thornton
SLCET-D
Fort Monmouth, NJ 07703-5000

US Army LABCOM (1)
ET&D Laboratory
ATTN: Mr. Owen Layden
SLCET-IB-H
Fort Monmouth, NJ 07703-5000

US Army LABCOM (1)
ET&D Laboratory
ATTN: Mr. Herbert Mette
SLCET-DT
Fort Monmouth, NJ 07703-5000

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

US Army LABCOM (1)
ET&D Laboratory
ATTN: Mr. E. Hakim
SLCET-RR
Fort Monmouth, NJ 07703-5000

US Army LABCOM (1)
ATTN: Richard Vitali
AMSLC-TD
2800 Powder Mill Road
Adelphi, MD 20783

US Army AVRADA (1)
ATTN: Joe DaSaro
SAVAA-S
Fort Monmouth, NJ 07703-5000

US Army CNVEO (2)
ATTN: John Pupich
AMSEL-NV-AC
Fort Belvoir, VA 22060-5677

US Army Center for Signal Warfare (1)
ATTN: AMSEL-RD-SW-OS
Dr. Royal Burkhardt
Vint Hill Farm Station
Warrenton, VA 22186-5100

US Army EW/RSTA Center (1)
ATTN: Mr. John Cervini
AMSEL-RD-EW-C
Fort Monmouth, NJ 07703-5000

US Army LABCOM (1)
ET&D Laboratory
ATTN: Dr. Dean Woo
SLCET-SI
Fort Monmouth, NJ 07703-5000

US Army AVRADA (2)
ATTN: Doug Johnson
SAVAA-I
Fort Monmouth, NJ 07703-5000

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

US Army Missile Command (1)
ATTN: Mr. Tilton Tippet
AMSMI-RD-AS-RA
Redsone Arsenal, AL 35898

US Army Missile Command (1)
ATTN: Mr. Robert Mitchell
AMSMI-RD-AS-OG
Redstone Arsenal, AL 35898

US Army Missile Command (1)
ATTN: Mr. Ted Peacher
AMSMI-RD-AS-IR
Redstone Arsenal, AL 35898

Center for Communications/ADP (1)
ATTN; Mr. Eugene Famolari, Dir
AMSEL-RD-COM-D
Fort Monmouth, NJ 07703-5000

US Army LABCOM (25)
ET&D Laboratory
ATTN: Vincent Organic
SLCET-IV
Fort Monmouth, NJ 07703-5000

US Army LABCOM (1)
SLCET-DT (M. Howard)
Fort Monmouth, NJ 07703-5000

CECOM R&D Technical Library (3)
AMSEL-IM-L (STINFO)
Fort Monmouth, NJ 07703-5000

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

NAVY

Navy VHSIC Office (5)
LCDR P. Gariano
SPAWAR 3213
Washington, DC 20363-5100

Naval Research Laboratory (1)
ATTN: Mr. Jim Alter
Code 5305
Washington, Dc 20375

Office of the Ass't Secretary of the Navy (1)
ATTN: Dr. A. E. Berman
The Pentagon, Room 5E-731
Washington, DC 20301

Naval Air Development Center (1)
ATTN: Mr. Paul Reimel
Code 4042
Warminster, PA 18974

Naval Surface Weapons Center (1)
ATTN: Dr. Harvey Alperin
White Oak, Code R45
Silver Spring, MD 20910

Naval Underwater System Center (1)
ATTN: Mr. John O'Sullivan
Code 3211
New London, CT 06320

Commanding Officer (1)
Naval Avionics Center
ATTN: Mr. Ron Jennings
Branch 964
6000 East 21st Street
Indianapolis, IN 46218

Naval Research Laboratory (15)
ATTN: Mr. J. P. Letellier
Code 5305
Washington, DC 20375

Naval Weapons Center (1)
ATTN: Mr. Vern Anderson
Code 33181
China Lake, CA 93555

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

Naval Air Systems Command (1)
ATTN: Mr. Steven Linder
Code 5143G
Washington, DC 20361

Naval Research Laboratory (1)
ATTN: Mr. Harold L. Hughes
Code 6816
Washington, DC 20375

Naval Weapons Support Center (1)
ATTN: Mr. William Brenner
Bldg. 38, Code 6062
Crane, IN 47522

Naval Air Systems Command (1)
ATTN: Mr. Richard Balestra
Code 5143E
Washington, DC 20361

SPAWAR (1)
ATTN: Ms. Diane C. Thornewell
Code 1232A
Washington, DC 20363

Naval Air Systems Command (1)
ATTN: Thomas Schlegel
Code 933
Washington, DC 20361

SPAWAR (1)
ATTN: Mr. T. Barry Hughes
Code PMW-151-1
Washington, DC 20363

Naval Research Laboratory (1)
ATTN: Mr. Joseph Killiany
Code 6813
Washington, DC 20375

Naval Ocean Systems Center (1)
ATTN: Dr. Isaac Lagnado
Code 5503
271 Catalina Blvd.
San Diego, CA 92152

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

Office of Naval Research (1)
ATTN: Dr. Clifford Lau
1030 East Green Street
Pasadena, CA 91106

Naval Research Laboratory (1)
ATTN: Dr. Martin C. Peckerar
Code 6804
Washington, DC 20375

Naval Sea Systems Command (1)
ATTN: Cmdr Bruce Gallemore
PMS 412
Washington, DC 20362

Naval Avionics Center (5)
ATTN: Mr. Jim Holtkamp
Code B834
6000 East 21st Street
Indianapolis, Indiana 46218

Naval Surface Weapons Center (1)
ATTN: Mr. Donald Becker
Code F-16
Dahlgren, VA 22448

Vela Associates (1)
ATTN: Albert Brodzinsky
4405 East/West Hwy
Suite 502
Bethesda, MD 20014

Eagle Research Group (1)
ATTN: Dr. John Davey
1925 North Lynn Street
Arlington, VA 22209

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

AIR FORCE

Mr. Darrell Barker (5)
AFWAL/AADE-1
Wright-Patterson AFB
Ohio 45433-6543

Mr. Robert Werner (25)
AFWAL/AADA VHSIC PO
Wright-Patterson AFB
Ohio 45433-6543

Lt. Col. John M. Borky (1)
ASD/TAS
Wright-Patterson AFB
Ohio 45433-6543

Mr. Jack Farrell (1)
RADC/RBRA
Griffiss AFB
New York, 13441

Mr. John Fenter (1)
AFWAL/MLTE
Wright-Patterson AFB
Ohio 45433-6543

Mr. Mason Friar (1)
AFWAL/AAAT
CSP Program
Wright-Patterson AFB
Ohio 45433-6543

Mr. J. Graniero (1)
RADC/DCCR
Griffiss AFB
New York, 13441

Lt. C. Grove (1)
AFSTC/YLD
Kirtland AFB
New Mexico 87117

Maj. Ed Herman (1)
ASD/YPRA
Wright-Patterson AFB
Ohio 45433-6543

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

Mr. Jerry Hudson (1)
AD/DLMI
Eglin AFB
Florida 32542

Mr. Richard Iliff (1)
WR-ALC/MMESC
Robins AFB
Georgia 31098

Maj. Robert Jones (1)
HQ AFSC/DLAE
Andrews AFB
Washington, DC 20334

Mr. George Konomos (1)
AFWAL/AAAS-3
Wright-Patterson AFB
Ohio 45433-6543

Mr. Tom Magner (1)
ASD/YYES
Wright-Patterson AFB
Ohio 45433-6543

Mr. Michael Mills (1)
ASD/AXS
Wright-Patterson AFB
Ohio 45433-6543

Mr. Jack Mineo (1)
RADC/OCT
Griffis AFB
New York, 13441

Dr. Dieter Schiller (1)
ASD/ENASC
Wright-Patterson AFB
Ohio 45433-6543

Dr. Walt Shedd (1)
RADC/ESR
Hanscom AFB
Massachusetts 01731

Lt. Marian Silverstein (1)
ESD/XRCT
Hanscom AFB
Massachusetts 01731

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

Lt. Col. Bob Stegmaier (5)
RADC/RBR
Griffiss AFB
New York, 13441

Mr. Ron S. Vokits (1)
ASD/AXTM
Wright-Patterson AFB
Ohio 45433-6543

Mr. Robert Williams (1)
AFWAL/AARM
Ultra Reliable Radar Program
Wright-Patterson AFB
Ohio 45433-6543

Mr. Nick Sramek (1)
Aerospace Corporation
P. O. Box 92957, M/S M4989
Los Angeles, CA 90009

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

DOD

Office of the Undersecretary of Defense (6)
Research & Advanced Technology
ATTN: VHSIC Program Director
The Pentagon, Room 3D-139
1211 Fern Street
Washington, DC 20301-3081

Palisades Institute for Research Services (25)
ATTN: L. Gallo
2011 So. Crystal Drive (Crystal Park 1)
Suite #307
Arlington, VA 22202

Defense Technical Information Center (2)
ATTN: DTIC-FDAC,
Bldg. 5, Cameron Station
Alexandria, VA 22304-6145

(NOTE TO CONTRACTOR: These two (2) copies for DTIC will be sent from STINFO, Fort Monmouth, NJ)

Defense Nuclear Agency (1)
ATTN: Lt. Cmdr Louis Cohen
HQ, DNA/RAEV
6801 Telegraph Road
Alexandria, VA 22310

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

INDUSTRY

ATT Technologies, Inc. (3)
ATTN: R. L. Jackson
P. O. Box 20046
Greensboro, NC 27420

RCA (1)
Advanced Technology Laboratories
ATTN: Mr. J. E. Saultz, ATL Bldg.
Moorestown Corporate Ct.., Rt. 38
Moorestown, NJ 08057

TRW, Inc. (5)
ATTN: Dr. Tom Zimmerman
1 Space Park, M5/1492
Redondo Beach, CA 90278

Honeywell, Inc. (1)
ATTN: Dallas Burns
MN14-4C40
12001 Hwy 55
Plymouth, MN 55441

Honeywell, Inc. (2)
ATTN: Mr. George Anderson
MN14-4E50
12001 Hwy 55
Plymouth, MN 55441

Honeywell, Inc. (5)
ATTN: Mr. D. Nielson
MN 14-4E50
12001 Hwy 55
Plymouth, MN 55441

Honeywell, Inc. (5)
ATTN: Dr. Jay Patel
P. O. Box 1361, MN65-3420
3660 Technology Dr.
Minneapolis MN 55440

Westinghouse Electric Corporation (5)
Defense Electronics Center
ATTN: David Sartorio
P. O. Box 1521-MS/5210
Baltimore, MD 21203

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

Westinghouse Electric Corporation (1)
ATTN: Dr. Gene Strull, Director
Advanced Technology Labs.
P. O. Box 746, M/S 5210
Baltimore MD 21203

Texas Instruments, Inc. (5)
VHSIC Program Office
ATTN: Mr. Jesse Wilson
M/S 96
P. O. Box 226015
Dallas, TX 75266

Hughes Aircraft Company (1)
Ground Systems Group
ATTN: Mr. Nate Newton
Bldg. 600/C245
P. O. Box 3310
Fullerton, CA 92634

Motorola Inc., Semiconductor Group (1)
ATTN: Mr. W. J. Kitchen
2200 West Broadway Road
Mesa, AZ 85202

Motorola, Inc. (1)
ATTN: Mr. Charles D. Meyer
3501 Ed Bluestein Blvd.
P.O. Box 6000
Austin, TX 78762

National Semiconductor (1)
ATTN: Mr. Jerry Streb
M/S D-3606
P. O. Box 58090
Santa Clara, CA 95052-8090

The Environmental Institute of Michigan (ERIM) (1)
ATTN: Dr. William Becher
P. O. Box 8618
Ann Arbor, MI 48107

Georgia Institute of Technology (1)
ATTN: Mr. Henry Owen
EES SEL/ESMD
P. O. Box 4203
Atlanta, GA 30332

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

Harris Corporation GASD (1)
ATTN: Mr. Lee R. Allain
P. O. Box 9400, Bldg. 63-15
Melbourne, FL 32902

Hewlett Packard (1)
ATTN: Mr. Bob Burmeister
1501 Page Mill Road., 1U
Palo Alto, CA 94304

Intermetrics, Inc. (1)
ATTN: Dr. Roger Lipsett
4733 Bethesda Ave., Suite 415
Bethesda, MD 20814

Litton Data Systems (1)
ATTN: Mr. Gary L. Mallaley
8000 Woodley Ave., M/S 44-37
Van Nuys, CA 91409

The MITRE Corporation (3)
ATTN: Mr. James R. Spurrier
Burlington Rd., M/S K224
Bedford, MA 01730

Raytheon Company (1)
ATTN: Mr. George Agule
Missile Systems Div., M/S M20-66
Hartwell road
Bedford, MA 01730

Research Triangle Institute (1)
ATTN: Mr. James B. Clary
Bldg. 5 - Digital Systems Section
P. O. Box 12194
Research Triangle Park, NC 27709

SAIC (1)
ATTN: Mr. Tim Geiger
1710 Goodridge Dr., T-12-1
McLean, VA 22102

Boeing Aerospace Company (1)
ATTN: T. A. Nicolino
P. O. Box 3999, M/S 8H-09
Seattle, WA 98124

November 1988

VHSIC Phase 2 Interoperability Standards Distribution List

E-Systems, Inc. (1)
ATTN: Mr. Charles Brodnax
Garland Div.
1200 S. Jupiter Rd.
P. O. Box 660023
Dallas, TX 75042

Hughes Aircraft Company (1)
Solid State Products Division
ATTN: Mr. Richard Belardi
Bldg 700/MS1250
500 Superior Avenue
Newport Beach, CA 92658-1420

IBM Federal Systems Division (1)
ATTN: Mr. Philip Johnson
9500 Godwin Drive
Manassas, VA 22110

Hughes Aircraft Company (1)
Hughes Microelectronics Center
ATTN: Dr. P. Hudson
6155 El Camino Real
Carlsbad, CA 92009

Hughes Aircraft Company (1)
ATTN: Lee Tower
Bldg R1, MS 5A27
2000 E. Imperial Highway
El Segundo, CA 90245

General Dynamics Convair Division
ATTN: Paul Hedke
5000 Kearney Villa Road
San Diego, CA 92138

Boeing Electronics, Inc. (1)
ATTN: Dr. Jerry Williams
Mail Stop 9J12
P.O. Box 2496
Seattle, WA 98124-6269

AT&T Bell Laboratories (1)
ATTN: Dennis Seals
1 Whippany Road
Whippany, NJ 07981

Ford Aerospace (1)
ATTN: R.L. Kopp
Ford Road
Newport Beach, CA 92658-8900